MASTERARBEIT | MASTER'S THESIS

Titel | Title

Vehicle Routing Problems with Backhauls – Implementation and Analysis using Large Neighborhood Search Algorithms

verfasst von | submitted by

Pauline Peter

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2024

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 915
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Betriebswirtschaft
Betreut von | Supervisor:	Ass.-Prof. Dr. Christian Tilk

Abstract

This thesis investigates the Vehicle Routing Problem with Backhauls (VRPB), a subset of the Capacitated Vehicle Routing Problem (CVRP), focusing on three variants: VRP with Clustered Backhauls (VRPCB), VRP with Divisible Pickup and Delivery (VRPDDP), and VRP with Simultaneous Pickup and Delivery (VRPSPD). The research aims to compare these variants in terms of their effectiveness in reducing transportation costs. To achieve this, the Large Neighborhood Search (LNS) algorithm was implemented, employing predefined destroy and repair operators to optimize initial route plans. Computational results, obtained after running the metaheuristic for 1,000,000 iterations, reveal that the LNS algorithm consistently improved solutions across all instances, with the VRPCB achieving the highest percentage improvements and the VRPSPD yielding the lowest final costs. Despite the challenges faced by the VRPDDP in reducing costs, adjustments to parameters such as Reinsertion Options led to further cost reductions. Overall, the study demonstrates the LNS algorithm's efficiency in minimizing transportation costs and highlights the ongoing significance of reverse logistics in modern supply chain management.

Abstrakt

Diese Arbeit untersucht das Vehicle Routing Problem with Backhauls (VRPB), eine Unterkategorie des Capacitated Vehicle Routing Problems (CVRP), und konzentriert sich auf drei Varianten: VRP with Clustered Backhauls (VRPCB), VRP with Divisible Pickup and Delivery (VRPDDP), und VRP with Simultaneous Pickup and Delivery (VRPSPD). Ziel der Untersuchung ist es, diese Varianten im Hinblick auf ihre Effektivität bei der Transportkostensenkung zu vergleichen. Um dies zu untersuchen, wurde ein Large Neighborhood Search (LNS) Algorithmus implementiert, der vordefinierte Destroy- und Repair Operatoren zur Optimierung der ursprünglichen Routenpläne verwendet. Die Berechnungsergebnisse, die nach der Ausführung der Metaheuristik für 1.000.000 Iterationen erzielt wurden, zeigen, dass der LNS-Algorithmus die Lösungen über alle Instanzen hinweg durchgängig verbesserte, wobei die VRPCB Variante die höchsten prozentualen Verbesserungen und die VRPSPD Variante die niedrigsten Endkosten erzielten. Trotz der Herausforderungen, mit denen die VRPDDP Variante bei der Kostensenkung konfrontiert war, führten Anpassungen von Parametern wie Reinsertion Options zu weiteren Kostensenkungen. Insgesamt zeigt die Studie die Effizienz des LNS-Algorithmus bei der Minimierung der Transportkosten und unterstreicht die anhaltende Bedeutung von Reverse Logistics im modernen Lieferkettenmanagement.

Table of Contents

I List of Figures

II List of Tables

1 Introduction

In the context of rising costs across multiple industries, optimizing delivery routes has emerged as a key strategy for companies to improve efficiency without incurring additional losses (Battarra, 2014). Reverse logistics (RL) plays a vital role in the optimization process. It involves activities essential to the efficient collection process, including the recycling and reuse of products (Sar & Ghadimi, 2023). This approach not only minimises costs but also reduces waste, improves environmental performance, and ensures social responsibility (Konstantakopoulos, 2022; Ayvaz, 2015). The growing emphasis on reverse logistics has impacted transport companies, requiring them to manage both delivery and pickup demand services efficiently. This dual role has transformed traditional one-way logistics into a more complex problem. This has not only caught the attention of industry practitioners but has also attracted considerable academic interest (Han & Ponce Cueto, 2015; Avci, 2015). In particular, researchers have focused on developing optimization techniques to meet the new challenges in route planning and scheduling posed by the increasing importance of RL. Many studies in the literature focus on aspects such as network design, economic and environmental performance, lot sizing, return forecasting, and vehicle routing.

Over the past two decades, *one-to-many* problems, such as the Vehicle Routing Problem with Backhauls (VRPB), have gained popularity, driven by the increasing emphasis of RL, e.g., in the grocery industry (Koç, 2018). Despite extensive research on VRPBs, a detailed analysis of the differences among its three variants is lacking. This thesis addresses this gap by investigating and comparing the three VRPB variants: the Vehicle Routing Problem with Clustered Backhauls (VRPCB), the Vehicle Routing Problem with Simultaneous Pickup and Delivery (VRPSPD), and the Vehicle Routing Problem with Divisible Pickup and Delivery (VRPDDP). The study compares these subcategories regarding reductions in transportation costs and examines how each variant differs based on predefined parameters through a computational study.

The remaining sections are structured as follows: This master thesis will begin with a literature review, focusing on VRPB and their variants, alongside heuristic and metaheuristic solution strategies. It will then outline the mathematical formulations and objective function specific to these VRPB variants in Section 3. Section 4 describes implementing of the Large Neighborhood Search (LNS) algorithm, detailing the initial solution setup and optimization techniques used.

A comprehensive analysis of computational results will assess the efficiency and effectiveness of these methods in Section 5. The thesis concludes with a summary of the findings.

I extend my sincere appreciation to the Centre of Logistics and Sustainability Analytics (CeLSA), particularly to Mr. Saïd Salhi and Mr. Gábor Nagy, for providing the input data used in this study[1]. This dataset was originally published in Salhi and Nagy (1999).

2 Literature Review

A common area of focus in logistics and distribution studies is the delivery of goods from a depot to customers, typically grouped under the term Vehicle Routing Problems (VRP). The classical VRP, a well-known combinatorial optimization problem, has been extensively studied due to its significant real-world applications (Tan & Yeh, 2021). The VRP was originally introduced by Dantzig and Ramser in 1959 as a truck dispatching problem. Since then, numerous variants have been developed and discussed.

The classic VRP aims to establish a series of cost-effective vehicle routes, ensuring that each customer is serviced exactly once, with all vehicles starting and ending at the depot(s). The routes are designed with specific constraints, which will be described in Section 3.2. One of these constraints guarantees e.g., that the vehicles' carrying capacities are not exceeded throughout the routes (Braekers et al., 2016). The Capacitated VRP (CVRP) as a variant extends the VRP to this exact point, including the fact that the vehicles have a certain capacity that is known beforehand. Within the broader VRP family, VRPB is a subclass of the CVRP, in which the customer set is divided into two subsets: linehaul and backhaul customers. Customers who receive goods from the depot are referred to as delivery or linehaul customers, while those who need a certain quantity to be picked up are known as pickup or backhaul customers (Wassan & Nagy, 2014). The VRPB deals with the logistics of managing both demands effectively by transporting each type of goods to and from the depot within a route (Mingozzi et al., 1999).

Routes are carefully planned to meet these individual customer requests while minimizing travel costs and adhering to vehicle capacity constraints (Souza et al., 2011). The VRPB in this

[1] available at https://research.kent.ac.uk/celsa/data-sets/#611-simultaneousfiles

master thesis denotes the symmetric version of the problem, in which traveling costs between each pair of customers are equal in both directions (Desaulniers et al., 2016).

The terminology used to describe the variants of the VRPB in academic literature can often be confusing, with different authors using the same term to refer to other variants (Nagy et al., 2015). Therefore, this master thesis adopts the definitions proposed by Parragh et al. (2008).

Subchapters 2.1 – 2.3 of this thesis will describe the three specific variants of the VRPB: VRPSPD, the VRPDDP, and the VRPCB. Each variant will be thoroughly analysed, highlighting its connections, similarities, and unique characteristics. This includes mathematical formulations, objective functions, and constraints, supplemented by illustrative graphics for clearer understanding. Additionally, a visual overview in the form of a table or graphic diagram may be provided to outline the core features and constraints of these three variants of the VRPB, further enriching the detailed analysis.

2.1 Vehicle Routing Problem with Clustered Backhauls (VRPCB)

The VRPCB is also recognized under several alternative names in the literature, including *VRP with Backhauling* (Granada-Echeverri et al., 2019; Konstantakopoulos et al., 2022; Wassan & Nagy, 2014), *Delivery-first, pickup-second VRPPD* (Nagy & Salhi, 2005), or *VRP with Backhaul and Linehaul* (Marinakis & Migdalas, 2007). As previously mentioned, this paper adopts the terminology proposed by Parragh et al. (2008), referring to the studied variant as the VRPCB.

According to Koç & Laporte (2018), the VRPCB was first introduced by Deif and Bodin in 1984. VRPCB customers can either be linehaul or backhaul customers (Avci & Topaloglu, 2015). Customers who require both services will be visited separately for each demand. The VRPCB operates under a precedence constraint, meaning that vehicles first serve customers with delivery needs, followed by those requiring pickups (Alfredo Tang Montané & Galvão, 2006; Battarra et al., 2014; Granada-Echeverri et al., 2019; Konstantakopoulos et al., 2022).

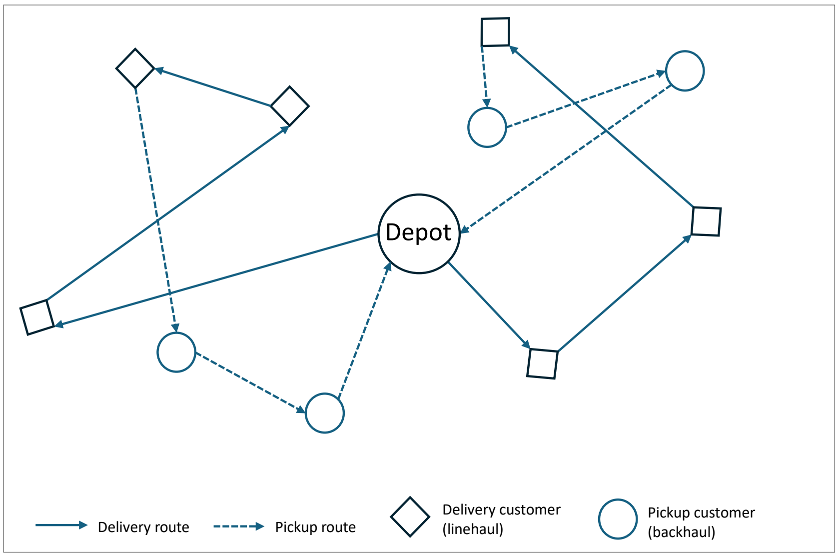

Figure 1: VRP with Clustered Backhauls

Figure 1 illustrates this variant graphically, an adaptation of Belloso et al. (2017). Squares indicate the delivery customer, while circles denote pickup customers. In the following part, including the computational study, the implementation of this variant is structured to ensure that backhaul customers are only added to the route once no further linehaul customers can feasibly be inserted at the considered position. However, insertion is still possible at another position, even if not all linehaul customers have been inserted yet.

The VRPCB approach is mainly driven by the logistical benefit of minimizing the need to rehandle goods along the route, an important consideration given that vehicles are often loaded from the rear (Belloso et al., 2017). Another advantage of this variant is that it helps in managing the vehicle's capacity more efficiently. Randomly visiting customers with pickup and delivery demands without observing the linehaul before the backhaul sequence can lead to exceeding the vehicle's capacity, rendering the route infeasible (Koç & Laporte, 2018). The VRPCB, therefore, allows for fluctuations in load throughout the route, increasing the likelihood of maintaining feasible routes (Nagy & Salhi, 2005).

Dinc Yalcın and Erginel (2015) have applied the VRPCB for the first time to the real-life data of a ceramics firm in Turkey. In their study, they successfully serviced 93 linehaul customers and 27 backhaul customers, effectively utilizing vehicle capacity and optimizing transportation costs.

2.2 Vehicle Routing Problem with Simultaneous Pickup and Delivery (VRPSPD)

The literature on the VRPSPD is significantly broader than for the VRPCB. In the late 1980s, Min (1989) introduced the VRPSPD, which was applied to distribute books across 22 libraries in Franklin County, Ohio, using the central library as a depot and employing two vehicles (Chentli et al., 2018; Nagy & Salhi, 2005; Sar & Ghadimi, 2023; Souza et al., 2011).

The key difference to other variants lies in the customer visits: in the VRPSPD, a customer with both pickup and delivery demands is served at one single visit, integrating both services simultaneously without separating the demands into categories of pickup and delivery (Alfredo Tang Montané & Galvão, 2006; Konstantakopoulos et al., 2022; Zachariadis & Kiranoudis, 2011). Figure 2 illustrates the variant structure by using the double-lined arrows to connect the nodes for synchronized services of pickup and delivery.

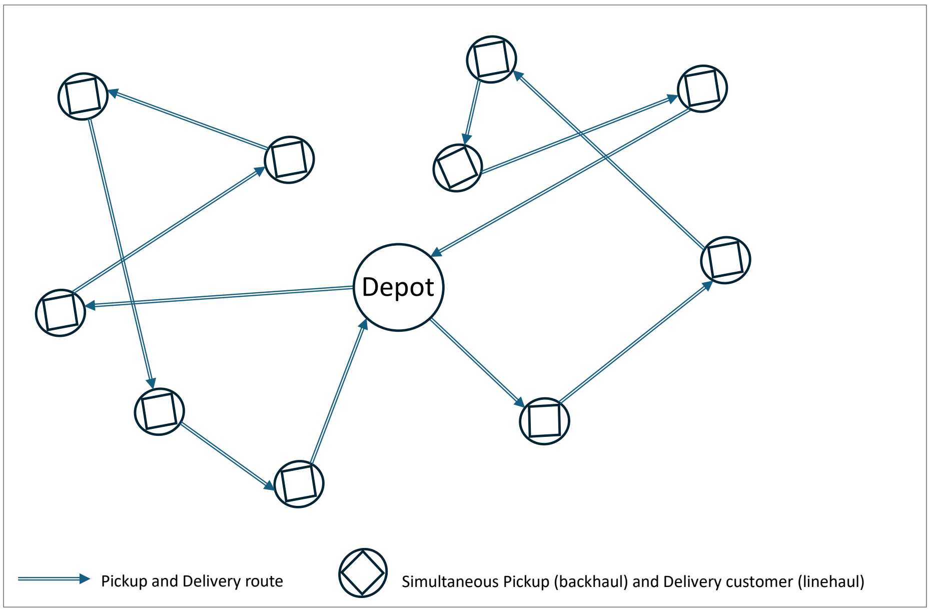

Figure 2: VRP with Simultaneous Pickup and Delivery

This integration significantly reduces transportation costs as it combines the delivery and pickup activities of a single customer, which is an advantage of this variant (Demirbilek, 2021). Additionally, the VRPSPD uses modelling frameworks similar to other problem types such as the VRPCB, simplifying its implementation (Nagy & Salhi, 2005). While the VRPSPD reduces customer expenses, it may lead to an increase in overall route lengths (Ropke & Pisinger, 2006).

The application is widespread, including industries such as soft drink companies where empty bottles need to be collected for return (Alfredo Tang Montané & Galvão, 2006; Chentli et al., 2018) and supermarkets that repurpose empty pallets for shipping (Avci & Topaloglu, 2015). Another environmentally significant application is found in foundries, where the VRPSPD is utilized to transport toxic sand used in moulding, which must be recycled rather than disposed of. This requires the optimization of vehicle routes for both the collection of used materials and the distribution of cleansed reusable materials (Dethloff, 2001). The broad applicability and essential advantages of the VRPSPD make it a flexible and cost-effective solution for various industries, demonstrating its significance in the field of logistics optimization.

2.3 Vehicle Routing Problem with Divisible Pickup and Delivery (VRPDDP)

In contrast to the VRPCB and VRPSPD, the VRPDDP does not require deliveries and pickups to follow a specific sequence, hence allowing them to occur in any order along a vehicle's route. Each customer requires either pickup and delivery, only pickup, or only delivery demand. Customers with both deliveries and pickups must be served separately, resulting in a split of one customer node into two nodes: one for the pickup demand visit and another for the delivery demand visit, using the same split structure in the VRPCB. The VRPDDP is sometimes referred to as Mixed Pickups and Deliveries (Mixed VRPPD) but indicates the same procedure (Nagy & Salhi, 2005).

One significant advantage of implementing VRPDDP is the possibility of adapting from the VRPSPD framework. This can be achieved by assigning a value of zero to the delivery demand of the pickup customer or pickup demand of a delivery customer, respectively. This modification does not substantially increase the complexity of the model. Moreover, the VRPDDP can be seen as a *relaxation* of VRPSPD, allowing for the possibility that each customer requiring both pickup and delivery services may be visited twice (Ropke & Pisinger, 2006).

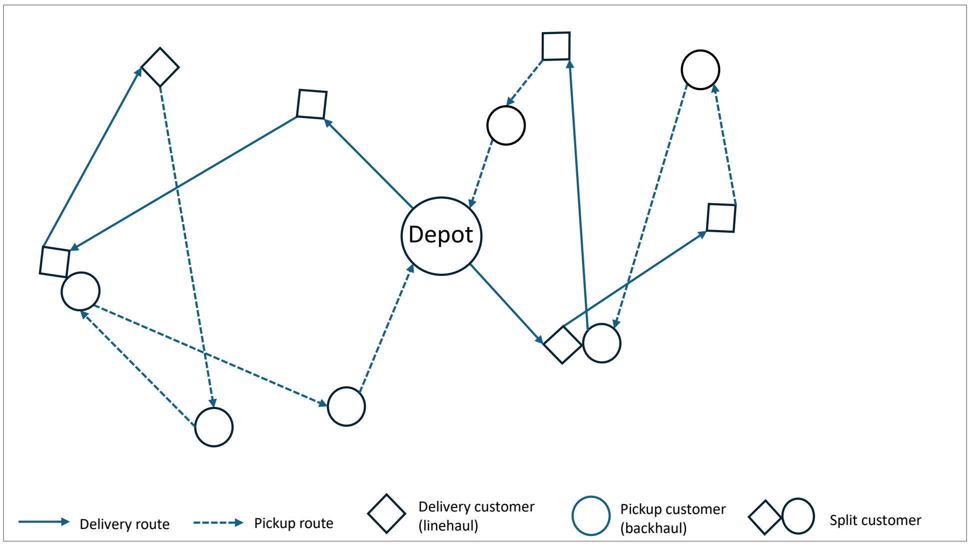

Figure 3: VRP with Divisible Pickup and Delivery

On one hand, this variant comes with a drawback as it doubles the number of customers, thus potentially extending the time needed to find a solution (Nagy et al., 2015). On the other hand, in many variations of the VRPDDP, routes that could initially be infeasible due to the simultaneous requirements for delivery and pickup loads might become feasible if multiple visits are permitted. This flexibility helps in managing the fluctuations in vehicle load more effectively (Wassan & Nagy, 2014).

The illustration in Figure 3 depicts the VRPDDP, where circles denote pickup customers and squares indicate those who require delivery services. Neighbouring signs show split customers, whereas arrows represent the direction of the vehicle routes.

The formation of vehicle routes in VRPDDP offers multiple strategic approaches. Golden, Baker, et al. (1985) have described a method where pickup customers are inserted into routes initially designed for delivery customers using a penalty factor to optimize the insertion. Extending the methods by Casco, Golden, and Wasil (1988), Salhi & Nagy (1999) have enhanced their insertion technique to allow for the clustering of backhauls. This paper utilizes Sequential Insertion heuristics, where the cost of inserting a customer at each potential position is calculated, aiming to place customers at positions where the insertion costs are minimized.

2.4 Literature regarding Heuristic and Metaheuristic solution approaches

Subsection 2.4 explores various heuristic solution methods used to solve different forms of VRPs. Over the past six decades exact algorithms, heuristics, and metaheuristics have been applied in operations research to address capacitated VRPs that are considered NP-hard problems (Konstantakopoulos et al., 2022; Lenstra & Kan, 1981). Heuristic and metaheuristic solution approaches, especially, have shown to be effective in solving a wide range of complex issues where exact algorithms cannot ensure an optimal solution (Cabrera-Guerrero et al., 2016). Heuristics typically involve an iterative procedure which improves a given feasible solution by exploring neighbouring solutions that better satisfy the objective function or other predefined solution attributes (Crainic & Toulouse, 2010). Metaheuristics, a more refined form of heuristics, enhance solution quality by integrating local search techniques and strategies to avoid being stuck in local minima (Gendreau & Potvin, 2010).

According to Konstantakopoulos (2022), among all types of introduced algorithms for the VRP and its variants in the literature from 2010 to 2020, metaheuristic algorithms are most commonly used. They appear within 252 papers, followed by heuristics, named in 122 papers. In contrast, exact algorithms like Branch-and-Bound and Branch-and-Price are less frequently implemented (32 papers) due to their limitations with input data. Specifically, these exact algorithms efficiently solve problems with smaller amount of nodes and become impractical as the data size increases (Konstantakopoulos et al., 2022; Laporte et al., 2014).

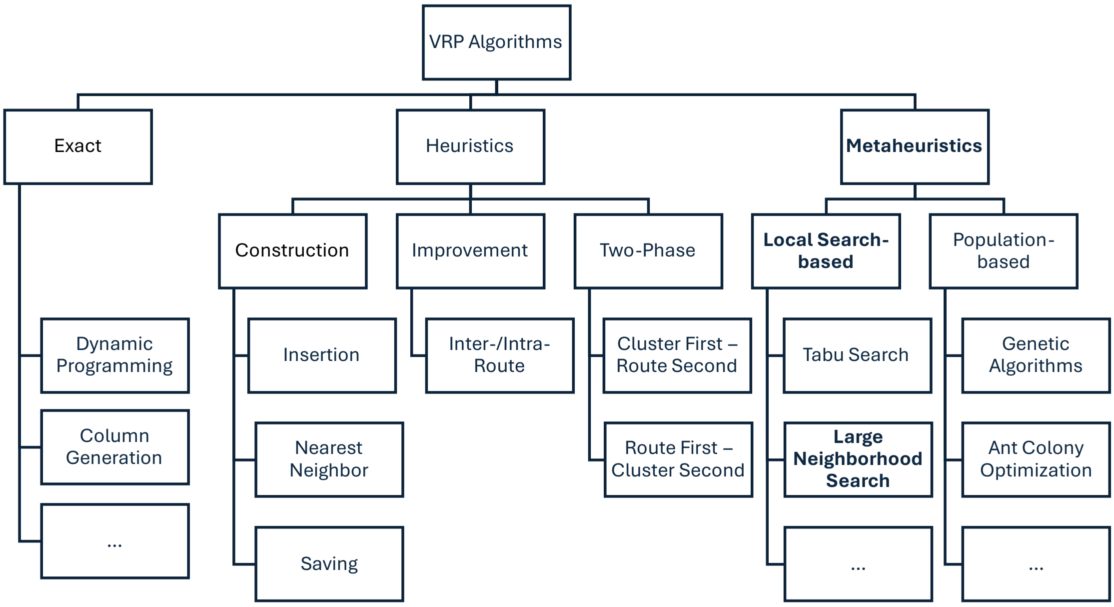

Figure 4: Classification of algorithms for the VRP (adapted from Konstantakopoulos et al., 2022)

Metaheuristics represent an advanced category of heuristics, including both population-based and local search-based methods, as shown in Figure 4. Population-based metaheuristics create new solutions from a set of existing ones by focusing on the global exploration of diverse solution sets. In contrast, local search-based metaheuristics focus on iteratively exploring the set of solutions by implementing minor, localized modifications within the local neighborhood (Mekamcha et al., 2021).

The classification of the LNS algorithm in the literature is a subject of debate. Some researchers classify it as a local search method due to its iterative improvement process within a neighborhood through destroy and repair operators (Konstantakopoulos et al., 2022). Conversely, other researchers suggest that the LNS algorithm should be categorized under Improvement heuristics as it iteratively modifies the current solution by performing local searches for better neighbouring solutions, similar to other r-exchange methods (Savelsbergh & Sol, 1995). For this study, we follow the argumentation of Konstantakopoulos et al. (2022) who classify the LNS algorithm as local search-based metaheuristics.

Route improvement heuristics, which belong to local search methods, are widely employed to explore solutions in the neighborhood to search for an improved solution regarding the objective. The fundamental concept of these heuristics involves relocating, exchanging, and reinserting customers or nodes to create an improved route plan (Bräysy & Gendreau, 2005). The LNS algorithm is a Local Search Algorithm often used in the literature alongside the Tabu Search (TS) and the Variable Neighborhood Search (VNS) due to its efficiency in handling large instances within a reasonable timeframe. The LNS operates by destroying the initial solution partly using removal operators and subsequently repairing it through repair operators, thereby exploring a large solution space (Konstantakopoulos et al., 2022).

Accordingly, this master's thesis will adopt a metaheuristic solution approach in the form of an LNS, as large-scale problems can be solved with such an algorithm.

3 Solution Framework

Chapter three outlines a uniform mathematical framework. To achieve this, the problem description is specified at first. Next, the notations of the mathematical formulation for all variants in this paper are presented and subsequently expanded to all variants of the VRPB. Finally, the solution method is outlined. The objective of this section is to provide clear definitions for the various problem types and the LNS algorithm.

All variations discussed are classified as NP-hard, extending the principles of the widely recognized Traveling Salesman Problem (TSP) (Garey, 1979).

3.1 Problem Description

The three subclasses of the classical VRP can be described on a complete and directed graph $G = (V, A)$ where V represents the set of all vertices and $A\ (i\ ,\ j)$ denotes the arc from vertex i to vertex j. Customers are categorized into two types: those requiring pickups, known as backhaul customers, and those needing deliveries, mainly as linehaul customers (Koç & Laporte, 2018; Toth & Vigo, 1997). The three variants can be classified as multi-vehicle pickup and delivery problems (Parragh et al., 2008).

Several assumptions have been made (Ropke & Pisinger, 2006):

1) Each vehicle is assigned to execute exactly one route.
2) The model contains a flexible maximum number of vehicles, each with the same capacity that cannot be exceeded.
3) Every customer has specific pickup, delivery, or both demands depending on the VRP variant.
4) All routes start and conclude at a single depot.
5) Transportation costs are calculated based on the distance between customers i and j, using the standard Euclidean metrics.

Furthermore, this study did not consider constraints related to time windows and travel duration limits.

3.2 Mathematical Formulations and Objective Function

The initial part of Section 3.2, including the notation and constraints, is applicable to all three previously discussed subclasses of the CVRP. Detailed explanations of specific adaptions respective to each variant will follow in the same section.

The framework for these capacity restricted multi-vehicle pickup and delivery issues employs a three-index VRP formulation, which was initially introduced by Cordeau et al. (2002) for the Vehicle Routing Problem with Time Windows (VRPTW). It was later adopted for the CVRP in this context, which is used in the following (Parragh et al., 2008).

The subsequent notation for the mathematical formulations is considered:

$V\ =\ \text{number of total vertices}$

$A\ =\ \text{number of all arcs}$

m = number of delivery vertices

n = number of pickup vertices

D = total number of delivery vertices (linehauls); $\mathrm{D} = \{\mathrm{p}+1, \dots, \mathrm{p}+\mathrm{d}\}$

P = total number of pickup vertices (backhauls); $\mathrm{P} = \{1, \dots, \mathrm{p}\}$

d_i = delivery demand at vertex i

p_i = pickup demand at vertex i

H = set of vehicles

C^h = capacity of vehicle h

c_{ij}^h = costs occuring when travelling from i to j with vehicle h

$$x_{ij}^h = \begin{cases} 1, \text{if vehicle h travels arc(i , j)} \\ \quad 0, \text{otherwise} \end{cases}$$

Q_i^h = load of vehicle h when leaving vertex i

The fundamental mathematical formulations are taken from existing literature regarding the CVRP, as they remain consistent across studies. The proposed problem is symmetric, meaning that $c_{ij} = c_{ji}$ and arc (i, j) is equivalent to arc (j, i). Additional constraints or adaptions specific to each variant are derived from designated sources: Battarra et al. (2014), Bianchessi & Righini (2007), and Ropke & Pisinger (2006) for the VRPSPD. Mosheiov (1998), Nagy et al. (2015) and Zachariadis & Kiranoudis (2011) for the VRPDDP and Granada-Echeverri et al. (2019), Nagy et al. (2015), Parragh et al. (2008), Ropke & Pisinger (2006) and Zachariadis & Kiranoudis (2011) for the VRPCB.

$$min \sum_{h \in H} \sum_{(i,j) \in A} c_{ij}^h \, x_{ij}^h \tag{1}$$

Subject to:

$$\sum_{h \in H} \sum_{j:(i,j) \in A} x_{ij}^h = 1 \qquad \forall i \in P \cup D, \tag{2}$$

$$\sum_{j:(0,j) \in A} x_{0j}^h = 1 \qquad \forall h \in H, \tag{3}$$

$$\sum_{i:(i,m+n+1)\in A} x_{i,m+n+1}^{h} = 1 \qquad \forall h \in H, \tag{4}$$

$$\sum_{i:(i,j)\in A} x_{ij}^{h} - \sum_{i:(j,i)\in A} x_{ji}^{h} = 0 \qquad \forall j \in P \cup D, h \in H, \tag{5}$$

$$\sum_{(i,j)\in A(S,\bar{S})} x_{i,j} \geq 1 \qquad \forall S \subseteq V \setminus \{m+n+1\}, S \neq \emptyset, \tag{6}$$

$$x_{ij}^{h} = 1 \rightarrow Q_j^h = Q_i^h + d_i + p_i \qquad \forall (i,j) \in A, h \in H, \tag{7}$$

$$max\,\{0, d_j + p_j\} \leq Q_i^h \leq min\,\{C^h, C^h + d_j + p_j\} \quad \forall i \in H, h \in H, \tag{8}$$

$$x_{ij}^{h} \in \{0,1\} \qquad \forall (i,j) \in A, h \in H, \tag{9}$$

The overall objective function (1) aims to minimize the total route costs over the vehicle fleet to satisfy both delivery and pickup demands. (2) ensures that each node is visited exactly once, while (3) guarantees that each vehicle h starts from the depot (node 0) and (4) that the vehicle h returns to the depot after completing its route. (5) deals with the well-known flow conservation constraint, meaning if a vehicle arrives at a node, it must also depart from it. Subtours are prevented with the subtour elimination constraint (6) while $\bar{S}$ is the subset of S, $A(S,\bar{S}) = \{(i,j) \in A: i \in S, j \notin S\}$. Constraints (7) and (8) refer to the load of the vehicle, ensuring that the load does not exceed the vehicle's capacity C^h after each pickup service and is correctly adjusted after each delivery service. (8) secures that the vehicle's load at each node respects both maximum capacity and minimum load requirements, possibly considering the current load as well as any additional pickup quantity. This is calculated by applying the Forward Labelling Algorithm, which will be explained in more detail in Section 4.1. Finally, (9) states that the variables are binary, indicating the presence or absence of a path between nodes i and j by vehicle h.

In addition to the classic constraints, each variant requires further requirements specified to their characteristics.

VRPCB

In the VRPCB, arcs that connects a backhaul to a linehaul customer are not allowed. This arrangement guarantees that every delivery demand customer *j* is initially visited by a vehicle for delivery purposes before picking up the demands at pickup customers i within each route. This sequence is enforced through the application of a precedence constraint (10) (Ropke & Pisinger, 2006):

$$x_{ij}^{h} = 0 \qquad \forall\, i \in P, j \in D, h \in H. \tag{10}$$

VRPSPD

On the other hand, the two variants, VRPDDP and VRPSPD, don't use such constraint (10), thus allowing linehaul and backhaul customers to be visited in any sequence (Zachariadis & Kiranoudis, 2011). Every customer must be served exactly once by a single vehicle to fulfil both pickup and delivery requirements at the same time. Additionally, unloading (linehaul) precedes loading (backhaul) the vehicle at these customer locations, ensuring an efficient operation (Ropke & Pisinger, 2006). No extra constraint is required for the VRPSPD.

VRPDDP

The third variant of the VRPB, the VRPDDP, can be directly transformed from the VRPSPD framework. When customers require pickup and delivery services, they are effectively split into two separate customer nodes within the model: one representing the pickup and the other node the delivery demand. Consequently, each customer can be visited twice, with the irrelevant demand (either pickup or delivery) set to zero for each visit. Constraint (10) along with the necessary implementation to consider each customer as two separate nodes, are sufficient for the VRPDDP without the need for additional constraints.

4 Solution Method

The Large Neighborhood Search algorithm (LNS) is chosen as the solution method. The methodology compromises two stages: the initial solution construction and the application of the LNS algorithm. The first stage is constructed using a Sequential Insertion Heuristic to create initial routes, which are then evaluated for feasibility and capacity using the Forward Labelling Algorithm. The LNS algorithm is subsequently applied in the second stage. The following subsections provide detailed information about the proposed algorithm.

4.1 Initial Solution Construction

For the LNS algorithm to be effective, an initial solution needs to be established. Several heuristic options are available to construct the initial solution: the Nearest Neighbour heuristics, the Savings heuristic, and the Sequential Insertion heuristics, to name a few, from which the last option was chosen for this paper. The pseudo code is given in Figure 5.

Initial Solution Construction

```
Create empty route_plan
while num_routes < num_vehicles do
    Create curr_route with depots
        if curr_route.size() == 2 then
            Select and insert seed customer with largest demand
        else
            for each uninserted customer and position do
                Calculate insertion cost and check feasibility
                If VRPSPD and customer is pickup customer then
                    Check if pos + 1 is not a delivery customer
                Check capacity constraints
            if feasible insertion found then
                Insert best customer into best position
            else
                Complete and add curr_route to route_plan
                Reset curr_route and curr_load
    Add last curr_route to route_plan
end while
Return initial_solution
```

Figure 5: Pseudo Code of Initial Construction Heuristic (adapted from Chentli, 2018)

In detail, the construction begins with an initially empty route and selects the first customer as the seed. The inserted seed customer is the one with the largest demand of either pickup or delivery. The selection depends on the respective variant and its constraints: the VRPCB prioritizes delivery before pickup, whereas this sequence is not required for the VRPSPD and VRPDDP (Chentli et al., 2018).

Continuing with the Sequential Insertion Heuristic for the remaining yet uninserted customers, each customer is considered successively, which is why it is termed sequential (Solomon, 1987). The algorithm determines each customer's best hence most cost efficient position by calculating the insertion costs for each possible position. It selects the option with the lowest additional cost, a method known as the Cheapest Insertion Heuristic

(Mitrović-Minić & Laporte, 2004). For the parameterized insertion costs, the parameters μ and λ are introduced. These parameters adjust the influence of the direct travelling costs and the depot-related travelling costs in the overall insertion cost calculation. Larger values of λ increase the preference of nodes located farther from the depot, while higher values of μ emphasize the importance of the cost associated with the removed edge. The values of both parameters μ and λ are chosen randomly using a random number generator.

The parameterized insertion costs $C(i,j)$ of the used algorithm are calculated as follows:

$$C(i,j) = c_{j-1,i} + c_{i,j} - \mu\, c_{j-1,j} - \lambda(c_{0,i} + c_{i,0}) \tag{11}$$

$c_{j-1,i}$ denotes the costs of the route from the previous node $j-1$ to the potential inserted node i, while $c_{i,j}$ denotes the costs of travelling from the potential node i to the next node in the route j, both coloured in green (Figure 6). Finally, $c_{j-1,j}$ calculates the route costs between both already existing and connected nodes that would be deleted (see red arrow in Figure 6) and $(c_{0,i} + c_{i,0})$ show the costs of the potential nodes to the depot. For each uninserted node, the insertion costs are calculated for all potential positions in the route, and the position with the lowest cost is selected.

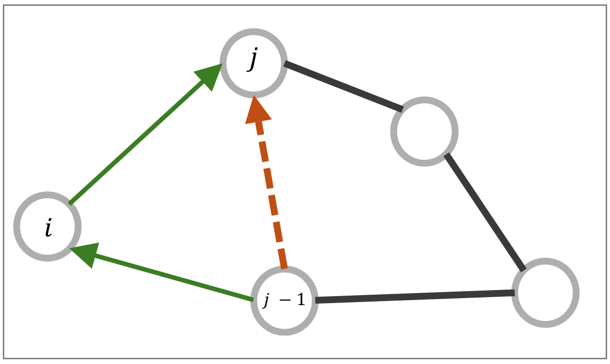

Figure 6: Illustration of parameterized insertion costs

Subsequently, a Forward Labelling Algorithm is applied for both variants, VRPSPD and VRPDDP, respectively, to check feasibility and capacity before inserting the considered node i into the current route at the chosen position pos. This algorithm verifies that the capacity of the vehicle h is not exceeded when reinserting each remaining node sequentially and adding its pickup and/or delivery demand to the current vehicle load.

The mathematical formulation for the maximum load definitions is presented below:

$M^r_{0,pos}$ = Total load until pos of node i in route r

$M^r_{pos,0}$ = Total load from pos of node i in route r until return to the depot

π^h_{i-1} = cummulated pickup demand from depot until potential inserted node

π^h_{i+1} = cummulated pickup demand from potential inserted node until depot

M^r_{i-1} = Maximum total load at node i for route r

M^r_{i+1} = Maximum total load from node i to depot

$$M^r_{i-1} = max\,\{\pi^h_{i-1}, M^r_{0,pos} + d_i\} \tag{12}$$

$$M^r_{i+1} = max\,\{\pi^h_{i+1}, M^r_{pos,0} + d_i\} \tag{13}$$

$$M^r_{i-1} \leq C^h \tag{14}$$

$$M^r_{i+1} \leq C^h \tag{15}$$

$$\pi^h_{i-1} \leq C^h \tag{16}$$

$$\pi^h_{i+1} \leq C^h \tag{17}$$

The Forward Labelling Algorithm calculates the cumulative pickup π^h_{i-1} of all inserted nodes starting from the depot to the proposed position pos of node i in route r. Next, the delivery demand d_i of the new potentially inserted node i is added to the total load $M^r_{0,pos}$ of vehicle h along route r, which includes all delivery demands of the travelled nodes up until the pos of node i. In constraint (12), both values π^h_{i-1} and $M^r_{0,pos} + d_i$ are compared, and the higher quantity is selected.

Taking π^h_{i-1} and adding all pickups of the remaining customers, results in π^h_{i+1} , representing the cumulated pickup demand until returning to the depot. By including the new nodes' demand of delivery as well as all other remaining delivery demands of all nodes after *pos* of node i, the load metrics are updated to $M^r_{pos,0}$. Comparing now both values π^h_{i+1} *and* $M^r_{pos,0} + d_i$ again in constraint (13) gives comparison M^r_{i+1}. It evaluates if inserting the node at this position keeps the greater of the total load or the maximum load within the vehicle's capacity limit of the vehicle (14) - (17). If the conditions are satisfied, this position is identified as feasible for insertion (Avci & Topaloglu, 2015).

This step is unnecessary for the VRPCB as the primary constraint is that delivery customers have to be visited before pickup customers. Once a customer has been inserted successfully, it cannot be revisited inside the construction of the initial solution. There is no repair function for infeasible solutions. Instead, the algorithm is reapplied iteratively until a feasible solution is identified (Avci & Topaloglu, 2015). Once the vehicle capacity is fully utilized, the route is completed and added to the route plan, after which a new route with a different vehicle is initiated. This process is repeated until all customers are inserted into routes whilst maintaining capacity and feasibility constraints. This procedure is applied separately to all three variants of the VRP. Subsequently, the LNS algorithm is applied to the routes by destroying and repairing them in order to improve the objective of minimizing total travel costs.

4.2 Large Neighborhood Search Algorithm

The LNS concept is a very general metaheuristic originally introduced by Shaw et al. (1998) and helps in finding high-quality solutions (Avci & Topaloglu, 2015; Pisinger & Ropke, 2010). It is defined as a combination of a destroy and repair operator, which allows the algorithm to avoid being stuck in local minima by making substantial changes to the current solution (Haddad et al., 2018). By defining a large neighborhood around the current solution, the LNS can explore a broader range of solutions, making it less likely to settle on non-optimal solutions (Lourenço et al., 2010). The destruction and repair of the initial solution hereby depends on the respective operators. A common extension to receive a more diverse set of potential improved solutions, proposed by Schrimpf et al. (2000) as Ruin and Recreated (R&R), is to define various destroy and/or repair operators rather than relying on just one of those. As a result, the neighborhood implicitly created by the combination of two operators is large, thus giving this metaheuristic its name (Pisinger & Ropke, 2010). This might let the success rate for an improved solution rise and gather insights on the performance of these different operators to the improved solution. Figure 7 displays the pseudo code of the LNS algorithm, which was adapted from Pisinger and Ropke (2010).

Large Neighborhood Search Algorithm

1: initial feasible solution x

2: $x^{best} = x$;

3: **repeat**

4: $x' = destroy(x)$;

5: $x' = repair\ (x')$;

6: **if** accept (x', x) **then**

7: $x = x'$;

8: **end if**

9: **if** $cost(x') < cost(x^{best})$ **then**

10: $x^{best} = x'$;

11: **end if**

12: **until** no. of iterations is reached

13: **return** x^{best}

Figure 7: Large Neighborhood Search Algorithm (adapted from Ropke & Pisinger, 2010)

The LNS algorithm starts by establishing an initial feasible solution x. This initial solution is typically built using a construction heuristic, such as the Sequential Insertion Heuristic, which helps to create a usable starting point for the search process. Once this initial solution is generated, it is set as the best found solution, x_{best} (line 2).

A predefined stopping condition, such as a set number of iterations, regulates the LNS search phase. During each iteration, the algorithm first enters the destroy phase, where a certain destroy operator removes a pre-defined number of customers from the current solution x , creating a new partial solution x' (line 4). The extent of this destruction is controlled by the destruction ratio parameter. In the repair phase, a repair operator is applied to reinsert the removed customers into x', forming a new complete solution (line 5). If the repaired solution meets all constraints, such as vehicle capacity and feasibility, it is considered for the next iteration (line 7).

Finally, the cost of the newly generated solution x' is compared to that of the best found solution x_{best} . If x' has lower cost, it replaces x_{best} (line 10). This iterative process continues until the chosen stopping condition is met and the best found solution x_{best} is returned as the final best solution (line 13).

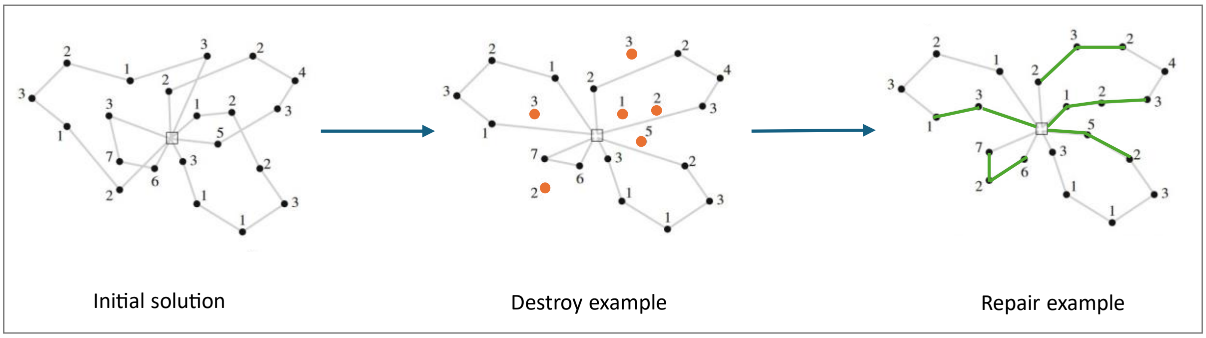

Figure 8: Remove - Repair movement during LNS algorithm (adopted from Ropke & Pisinger (2010))

The illustration in Figure 8, modified from Ropke & Pisinger (2010), graphically demonstrates the process of the LNS algorithm. The first diagram depicts the initial solution. The image in the middle shows the removal of specific nodes (in orange) to generate a partial solution, in which the removed nodes are then reinserted (in green) to obtain an optimized route (Repair example).

In this paper, the LNS heuristic, including the destroy and repair algorithms, is applied to each variant separately and was repeatedly run for multiple iterations (10,000, 100,000 and 1,000,000 iterations). Chosen destroy operators are Random Removal, Smallest Demand Removal and Largest Demand Removal. Brief descriptions of each algorithm are provided below:

1) **Random Removal:** The most straightforward removal heuristic, random removal, selects N number of customers at random and removes them from the solution. This has the effect of diversifying the search (Schrimpf et al., 2000).
2) **Largest Demand Removal:** Removal of N nodes with the largest demand.
3) **Smallest Demand Removal**: Removal of N nodes with the lowest demands.

A destroy parameter is randomly selected with equal likelihood, based on each iteration's number of customers removed. The choice of such a destruction parameter is crucial in the LNS algorithm, as it significantly impacts its effectiveness (Shaw et al., 1998). If the destruction parameter is set too low on the one hand, the search will likely become trapped in local optima and cannot explore the solution space. On the other hand, if the parameter is too high, the algorithm risks turning into a random search, thereby losing its structured approach and efficiency (Pisinger & Ropke, 2010). After carefully calibrating the destruction parameter, a destruction ratio between 30 - 60 % is randomly selected.

To rebuild the destroyed routes and reinsert the removed customers, the repair method is applied. Various algorithms, such as k-Regret heuristics, Feasibility-Oriented Insertion heuristics or the Greedy Insertion Heuristic can be employed to reinsert the customers (Lei et al., 2011).

The Greedy Insertion heuristics has been chosen for this implementation to find an improved solution regarding the objective of minimizing the total travelled costs. This heuristic is an effective choice for this paper's implementation as it focuses on making locally optimal decisions at each step, which can lead to a rapid and significant reduction in total travelled costs. It involves sequentially inserting customers into the routes based on a predetermined ordering. The sorting criterion is selected according to descending demand order, ascending demand order, or by randomly arranging the removed customer nodes. For each customer, the insertion position is determined based on the focus of minimizing the weighted sum of transportation costs within the route plan. Hence, this approach is considered a „greedy" heuristic (Potvin et al., 1996). Identical to the Initial Construction Heuristic, costs were calculated once again using the parameterized insertion costs calculation, including the previously described parameters μ and λ. This involves computing the difference between the costs of the inserted arcs and the costs of the deleted arcs (Bianchessi & Righini, 2007). In this approach, rather than selecting only the optimal cost, the five lowest costs are considered. From these five costs, one is randomly selected. This parameter, called Reinsertion Options, was implemented to introduce greater variability and randomness. Section 5.3 will give results on the effect of tuning this parameter.

To check the satisfaction of the capacity constraint, the Forward Labelling Algorithm is applied for the variants VRPSPD and VRPDDP. Again, the Forward Labelling Algorithm is not applied for the VRPCB for the same reasons described in 4.1. When both feasibility and capacity are proven, the customer is reinserted in its minimum cost position. If no feasible route is available, a new route is opened. Once all removed customers have been reinserted and the route plan has been repaired completely, the total cost of the current route plan is compared to the total cost of the last best solution. The last best solution could either be the initial solution from the first destroy-repair movement or the last best solution from the past iterations, depending on which one yields the lower cost. A stochastic acceptance criterion allows for non-improving solutions to be accepted if they meet specific criteria, enhancing the algorithm's ability to explore the solution space and escape local minima. For instance, slightly worse solutions may be accepted with a certain probability if their costs are within the accepted probability. Accordingly, in this thesis, if the solution is up to 10 % worse in terms of costs than the costs of the previous best solution x_{best} over all iterations and is better than the current best solution of the last iteration, the new best solution of the route plan is accepted and set as the best solution for the next iteration. Otherwise, the new solution is rejected, the last best solution is still valid, and a new destroy-repair movement begins. Once all destroy and repair iterations are completed, the best solution with respect to the objective function is identified, ideally resulting in reduced travel costs.

5 Computational Results

This chapter presents the results of applying the proposed metaheuristic to various test instances provided by Nagy and Salhi (1999). The initial section offers a detailed description of these instances, including a comparison of computational time. This is followed by a comparative analysis of the three variants VRPCB, VRPDDP and VRPSPD including summaries of the test results. It also highlights their differences, specific characteristics, and the effects of parameter tuning. Finally, the efficiency of the different destroy operators Destroy Random, Destroy Smallest Demand, and Destroy Largest Demand is evaluated and compared.

5.1 Explanation and Characteristics of Data Set

The Large Neighborhood Search heuristic, provided by Nagy and Salhi (1999), was tested on 28 single depot instances (consisting of 14 sets each of type X and Y). The data set includes customers with individual customer IDs, X- and Y- coordinates, as well as respective pickup and/or delivery demands. Particularly, in all type Y sets, both pickup and delivery demands are swapped relative to the corresponding X set. The depot is listed with zero demand and vehicles hold a fixed maximum capacity ranging from 140 to 200 units. In contrast, the overall number of vehicles remains variable with no upper bound restriction. This study will not consider both values of maximum distance and drop time mentioned in each instance. The data set from Nagy and Salhi (1999) has often been referred to in literature for various successful algorithm applications to multiple variants of the VRPB (Avci & Topaloglu, 2015; Ropke & Pisinger, 2006). During the implementation, it had been observed that the original values of the provided instances did not lead to successful outcomes when running the VRPDDP. The complexity introduced by splitting customers in this variant reduced solution improvements of the initial solution across multiple attempts of 100, 1,000, 10,000, 100,000 and 1,000,000 iterations.

Consequently, some modifications have been made to the data set, limiting the number of customers per instance set to a maximum of 85 customers instead of a prior maximum of 199 customers. The updated data set now ranges from 40 to 85 customers. To ensure consistency regarding the accurate comparison among the variants, the adapted data set was utilized for all three variants. This change made it possible to achieve effective results for each set of instances.

Instance	vehicle capacity	n	D	P
CMT1X	160	50	465	314
CMT1Y	160	50	343	436
CMT2X	140	75	818	551
CMT2Y	140	75	710	659
CMT3X	200	40	318	247
CMT3Y	200	58	364	444
CMT4X	200	50	369	355
CMT4Y	200	70	464	546
CMT5X	200	73	563	496
CMT5Y	200	85	616	629
CMT6X	160	50	465	314
CMT6Y	160	50	343	436
CMT7X	140	75	818	551
CMT7Y	140	75	710	659
CMT8X	200	44	337	264
CMT8Y	200	63	413	471
CMT9X	200	54	404	371
CMT9Y	200	61	398	457
CMT10X	200	72	560	490
CMT10Y	200	74	502	565
CMT11X	200	71	360	484
CMT11Y	200	65	360	404
CMT12X	200	82	667	804
CMT12Y	200	69	492	529
CMT13X	200	64	302	445
CMT13Y	200	82	466	518
CMT14X	200	69	547	674
CMT14Y	200	75	522	539

n: number of customers; D: Delivery demand total; P: Pickup demand total

Table 1: Instance set characteristics

Table 1 illustrates the revised specific instance characteristics, including instance names, vehicle capacity, number of customers *n*, as well as delivery and pickup demand of all customers in total. Throughout the data, integer distances are used.

The implementation of the LNS algorithm for the three VRPB variants described in Section 4.2 was developed in Visual Studio (Version 2022) using the programming language C++. The code was performed on an Acer Swift SF314-51, equipped with an Intel(R) Core(TM) i5-7200U CPU at 2.50GHz, 2.70 GHz. Each code run of the chosen variant and number of iterations was executed five times and then averaged to obtain mean results. This approach was employed to verify the code's accuracy and to increase variation for subsequent comparison and analysis in this thesis. Only table 5, Figure 10 and Figure 11 focus on not averaged solution results.

Instance	n	VRPCB	VRPDDP	VRPSPD
CMT1X	50	36.61	175.67	53.51
CMT1Y	50	36.84	189.81	61.82
CMT2X	75	80.35	359.59	99.76
CMT2Y	75	**80.62**	372.90	102.88
CMT3X	40	21.18	124.50	39.80
CMT3Y	58	35.71	274.01	84.23
CMT4X	50	28.29	219.46	64.14
CMT4Y	70	50.34	440.74	111.09
CMT5X	73	55.10	440.58	106.76
CMT5Y	85	71.11	599.80	**150.02**
CMT6X	50	34.38	210.63	54.02
CMT6Y	50	33.86	237.22	61.53
CMT7X	75	68.26	429.81	100.57
CMT7Y	75	67.28	398.82	102.96
CMT8X	44	24.05	153.69	44.63
CMT8Y	63	41.71	**761.23**	95.01
CMT9X	54	32.35	216.81	68.10
CMT9Y	61	39.89	293.11	89.86
CMT10X	72	52.99	378.55	108.73
CMT10Y	74	56.31	425.86	121.47
CMT11X	71	46.09	334.56	121.78
CMT11Y	65	37.73	314.71	104.88
CMT12X	82	74.56	528.63	133.95
CMT12Y	69	37.21	315.77	72.85
CMT13X	64	38.36	339.65	102.99
CMT13Y	82	59.08	597.10	149.08
CMT14X	69	56.95	444.12	101.61
CMT14Y	75	39.49	351.11	68.84
$\sum$		1,336.69	9,928.43	2,576.89

Table 2: Average CPU time for 1,000,000 iterations (in seconds)

The average Central Processing Unit time (CPU) for 1,000,000 iterations of the LNS algorithm took 230.7 minutes in total for all variants, 22.28 minutes for the VRPCB, 165.47 minutes for the VRPDDP and 42.95 minutes for the VRPSPD. Table 2 displays the computational time required to solve each variant for the respective instance set. The numbers highlighted in bold indicate the longest durations for each variant. Primarily resulting from the customer split extending the algorithm's computational run time, the heuristic for the VRPDDP variant demanded the most time. Especially, instance set CMT8Y took the longest to solve despite containing 63 customers, and thereby only approximately ¾ of the maximum number of customers. The total CPU time remains relatively consistent across all iterations, which makes it predictable and intuitive. As the number of iterations increases, the computational time needed correspondingly rises. Thus, more iterations result in longer processing time but yield better solutions (Resende & Ribeiro, 2010).

5.2 Comparison of variants

The results of the LNS algorithm, applied to each of the variants VRPCB, VRPDDP and VRPSPD, are summarized in Table 3. The sign Δ symbolizes the relative improvement in percent based on the initial solution costs by calculating $\frac{(I_x - F_x)}{I_x}\, x\, 100$. Positive values indicate a reduction in costs (improvement), while negative values would indicate an increase (decline). Again, the average of five runs is presented as the final solution result. The initial solution costs, denoted as I_x, were determined using the Sequential Insertion Heuristic during the initial solution construction phase, which is explained in detail in Section 4.1. These costs are deterministic, leading to unchanging initial solutions across all runs. The initial solution for variant VRPSPD, I_{SPD}, consistently displays the lowest costs compared to the other variants among all instance sets.

Overall, improvements in cost reduction were observed in all variants and instance sets, which is promising. For the VRPCB, optimizations resulted in improvements ranging from 17.06% to 51.34%, and results for the VRPDDP from 17.01% to 46.56%. While both variants yield similar results, the VRPSPD, in contrast, observed relatively less reductions between 11.79% and 33.78%. The table highlights the best and worst instances for each variant in bold, illustrating that even the least successful run demonstrated a significant enhancement, with a minimum improvement of 11.79% in the VRPSPD variant. Mainly, instances with 62 - 82 customers show good improvements. Thus, it is evident that the LNS algorithm has effectively enhanced the performance across all variants.

		VRPCB			VRPDDP			VRPSPD		
Instance	n	I_{CB}	F_{CB}	Δ	I_{DDP}	F_{DDP}	Δ	I_{SPD}	F_{SPD}	Δ
CMT1X	50	1142	804	29.63%	1226	828	32.45%	836	632	24.38%
CMT1Y	50	1310	811	38.08%	1193	804	32.61%	792	608	23.23%
CMT2X	75	2058	1213	41.04%	1996	1347	32.52%	1251	985	21.28%
CMT2Y	75	2045	1191	41.77%	1812	1264	30.23%	1223	956	21.80%
CMT3X	40	1071	741	30.77%	844	700	**17.01%**	647	543	16.14%
CMT3Y	58	1491	913	38.74%	1308	943	27.91%	874	698	20.16%
CMT4X	50	1212	832	31.35%	1211	836	30.95%	862	653	24.20%
CMT4Y	70	1699	1114	34.41%	1666	1169	29.82%	1008	858	14.86%
CMT5X	73	1662	1138	31.54%	1586	1196	24.62%	1114	923	17.18%
CMT5Y	85	2087	1235	40.84%	1709	1304	23.67%	1327	964	27.39%
CMT6X	50	1142	804	29.63%	1226	822	32.99%	836	638	23.66%
CMT6Y	50	1310	809	38.24%	1193	810	32.12%	792	611	22.80%
CMT7X	75	2058	1216	40.89%	1996	1342	32.76%	1251	1002	19.92%
CMT7Y	75	2045	1185	42.04%	1812	1259	30.51%	1223	962	21.31%
CMT8X	44	1129	770	31.78%	1153	734	36.31%	730	575	21.21%
CMT8Y	63	1555	959	38.32%	1402	1000	28.64%	843	739	12.29%
CMT9X	54	1275	880	31.00%	1296	893	31.06%	888	687	22.68%
CMT9Y	61	1447	935	35.36%	1262	961	23.88%	826	729	**11.79%**
CMT10X	72	1706	1135	33.47%	1615	1198	25.85%	1117	909	18.64%
CMT10Y	74	1702	1152	32.33%	1903	1202	36.82%	1069	892	16.52%
CMT11X	71	1382	945	31.65%	1753	1066	39.21%	1088	953	12.43%
CMT11Y	65	974	808	**17.06%**	1205	911	24.37%	955	735	23.06%
CMT12X	82	1595	878	44.94%	1530	947	38.08%	1046	760	27.30%
CMT12Y	69	1047	**533**	49.11%	928	**597**	35.65%	614	**470**	23.42%
CMT13X	64	1259	825	34.49%	1464	925	36.82%	940	809	13.91%
CMT13Y	82	1672	1106	33.85%	2178	1164	**46.56%**	1352	973	28.03%
CMT14X	69	1362	663	**51.34%**	1244	717	42.36%	827	548	**33.78%**
CMT14Y	75	1127	555	50.75%	1037	614	40.81%	631	501	20.57%
Average		1484	934	37.08%	1455	984	32.38%	963	761	20.95%

n: number of customers, I: initial solution cost, F: final solution cost, Δ: (I-F)/I improvement in %, run: 1,000,000 iterations

Table 3: Comparison of VRPCB, VRPDDP and VRPSPD results based on solution costs

When evaluating the average improvements over all instance sets, VRPCB leads with an average improvement of 37.08%, followed by the VRPDDP at 32.38% and the VRPSPD with 20.95%, indicating that, on average, the VRPCB outperforms the other variants. This superior performance could be credited to the specific constraint of the VRPCB that requires visiting delivery customers before pickup customers. This constraint, while present, may be weaker compared to the constraints of, e.g., the VRPSPD. From these observations, it can be concluded that the VRPCB consistently delivers the best results in terms of overall cost reduction.

However, a focused analysis of the final solution costs $\mathrm{F_X}$ only reveals that the VRPSPD surpasses the other variants in this specific measure. The final costs throughout the variants were notably diverse, with VRPCB achieving the lowest final cost F_{CB} of 533 units, VRPDDP reaching down to F_{DDP} of 597 units, and VRPSPD presenting the lowest observed costs F_{SPD} of 470 units. All three results were observed in instance CMT12Y, which involved 69 customers. It needs to be said that this instance overall starts with a relatively low initial solution cost in comparison to all other instances. Figure 9 presents a graphical comparison of the final cost solutions for the three variants. The x-axis displays the number of customers n, arranged in the same order as the instances.

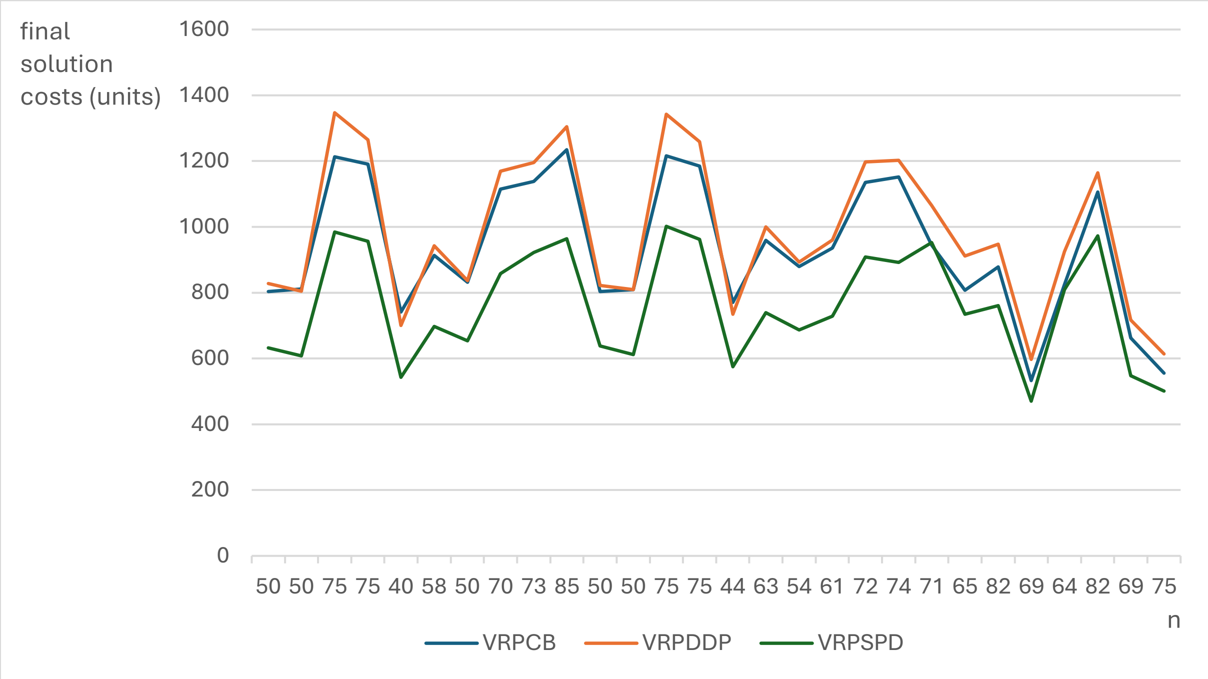

Figure 9: Comparison of VRPCB, VRPDDP and VRPSPD based on final solution costs

The VRPSPD variant (green line) steadily shows the lowest final solution costs across most instances (except in instance CMT11X), emphasizing the efficiency of the LNS algorithm implementation. The VRPCB (blue line) and VRPDDP (orange line) also demonstrate significant cost reductions but generally exhibit higher final costs compared to VRPSPD.

Comparison of final solution costs of VRPDDP vs. VRPSPD & VRPCB

Due to the more challenging constraints of the VRPCB (backhaul before linehaul customer visit) and the VRPSPD (backhaul and linehaul customer visit simultaneously), one would expect both variants to have more problems in decreasing the costs than the VRPDDP. Due to the relaxation of the VRPDDP, which allows the split backhaul and linehaul customers to visit separately, there should be more freedom in the algorithm to choose the cheapest position for both customers in the route plan. Contrary to this expectation, the F_{DDP} of the VRPDDP can in no instance beat the final solution cost F_{CB} of VRPCB and F_{SPD} of VRPSPD, as shown in Table 3.

One reason could be that the VRPDDP starts with an initial solution of the route plan where split customers are inserted at their best position, which does not necessarily mean they are placed next to each other. After removal, the split customer pair is also removed with a specific predefined removal ratio of 75%. In the repair method, all removed customers are reinserted, but not all pairs are reinserted next to each other. Debugging the code showed that only 7 out of 10 customer pairs were reinserted next to each other. This lack of consistent closeness could lead to higher final solution costs for the VRPDDP compared to the other variants. Another explanation could be that this happens due to the implementation where not all customer pairs are destroyed together, but only with a 75% chance. These remaining, non-removed customer parts could lead to inefficiencies, as the algorithm might not be able to optimize their positions as effectively as it could if all pairs were consistently destroyed and repaired as one. These factors likely add to the VRPDDP's higher final solution costs than the VRPCB and VRPSPD. A detailed comparison of paired customers on this topic is presented in Section 5.3.

Comparison of final solution costs of VRPCB vs. VRPSPD

From a practical perspective, the VRPSPD variant appears less complex due to its pickup and delivery service at the same stop. This might be a reason for the lower costs relative to the VRPSPD (Table 3). By forcing a delivery to precede a pickup customer node, the VRPCB constraint limits the possible routes the vehicle can traverse, leading to extended travel distances. Given the added complexity of these precedence constraints, the VRPCB's constraints can be considered harder. Therefore, it is understandable why the VRPSPD might achieve greater cost reductions compared to the VRPCB. Subsequently, the VRPCB shows significant improvements when comparing the initial solution with the final solution (% of improvement), while the VRPSPD shows the best improvements when comparing the final costs among all variants. To further explore how iteration counts affect the solution quality, Table 4 illustrates detailed iteration-specific data across various sets. Iteration sets range from 10,000, 100,000 to 1,000,000 iterations to run the LNS algorithm. The comparison helps in understanding how repeating the process optimizes the results. The table clearly indicates that the probability of discovering optimal solutions increases significantly with the number of iterations. The best final solution costs F_X are constantly found during the 1,000,000 iteration runs throughout all instances. This pattern underlines the reason for focusing the analysis primarily on results from the 1,000,000 iterations. Interestingly, despite setting a high iteration limit, optimal solutions often tend to appear around the 500,000 iteration mark well before reaching the maximum iteration count. As the algorithm runs longer, it may repeatedly find solutions close to the best solutions already discovered. This observation is consistent for both other iteration sets as well. This indicates that a higher iteration limit may not always translate to significantly better solutions and emphasizes the principle of diminishing returns.

		VRPCB						VRPDDP						VRPSPD					
		10000		100000		1.000.000		10000		100000		1.000.000		10000		100000		1.000.000	
Instance	n	iter	F_{CB}	iter	F_{CB}	iter	F_{CB}	iter	F_{DDP}	iter	F_{DDP}	iter	F_{DDP}	iter	F_{SPD}	iter	F_{SPD}	iter	F_{SPD}
CMT1X	50	4,353	845	71,214	822	728,314	804	6,074	890	41,955	860	469,146	828	6,040	709	48,160	677	614,462	632
CMT1Y	50	6,020	845	48,599	821	548,299	811	6,524	903	60,516	860	474,632	804	5,020	672	73,705	644	589,396	608
CMT2X	75	6,392	1,298	66,069	1,262	486,823	1,213	6,485	1,442	64,686	1,384	582,853	1,347	5,587	1,082	79,107	1,034	710,445	985
CMT2Y	75	7,177	1,285	78,783	1,217	800,055	1,191	1,693	1,624	65,208	1,322	665,040	1,264	5,448	1,035	60,811	993	672,353	956
CMT3X	40	6,643	771	52,991	747	633,974	741	5,742	841	56,189	741	516,659	700	5,199	624	58,895	574	481,027	543
CMT3Y	58	4,769	958	50,555	928	546,609	913	5,244	1,034	64,650	994	700,646	943	7,479	773	70,908	729	607,988	698
CMT4X	50	7,821	867	63,316	849	327,327	832	5,257	939	66,299	873	619,996	836	1,797	711	44,942	684	503,089	653
CMT4Y	70	7,639	1,170	59,101	1,142	537,660	1,114	4,670	1,280	65,125	1,207	612,030	1,169	4,525	921	40,156	895	603,316	858
CMT5X	73	7,634	1,199	46,244	1,162	667,234	1,138	7,173	1,362	54,243	1,234	477,217	1,196	6,370	1,003	39,188	925	715,112	923
CMT5Y	85	3,415	1,319	55,680	1,271	723,672	1,235	5,431	1,473	57,123	1,371	702,649	1,304	4,712	1,056	64,556	1,007	587,523	964
CMT6X	50	6,907	850	65,228	826	676,213	804	6,014	918	47,910	859	530,502	822	6,807	685	42,608	688	386,439	638
CMT6Y	50	5,774	846	51,811	821	491,507	809	5,097	912	66,854	853	607,817	810	6,543	669	50,940	625	681,393	611
CMT7X	75	8,618	1,279	60,031	1,263	452,133	1,216	6,508	1,446	49,327	1,394	284,913	1,342	7,594	1,064	45,955	1,031	342,632	1,002
CMT7Y	75	3,955	1,249	61,854	1,229	444,759	1,185	4,165	1,574	59,119	1,318	442,216	1,259	4,647	1,007	66,190	997	593,295	962
CMT8X	44	4,855	800	33,829	785	638,274	770	2,950	812	42,167	789	439,463	734	4,184	636	53,885	618	534,040	575
CMT8Y	63	4,958	1,035	58,778	981	606,495	959	6,757	1,167	64,548	1,061	717,724	1,000	4,439	813	86,451	784	739,044	739
CMT9X	54	6,138	933	71,603	891	626,504	880	6,743	987	54,633	936	465,916	893	4,521	767	39,172	717	520,787	687
CMT9Y	61	6,159	994	64,669	955	669,874	935	2,531	1,186	51,904	1,014	660,456	961	5,324	801	45,923	754	590,253	729
CMT10X	72	6,539	1,214	71,449	1,161	758,628	1,135	5,958	1,361	57,756	1,250	680,227	1,198	6,000	985	41,216	933	381,294	909
CMT10Y	74	6,211	1,231	62,155	1,180	469,282	1,152	3,622	1,311	50,681	1,258	707,326	1,202	3,555	978	51,607	939	389,042	892
CMT11X	71	7,392	1,023	59,302	959	573,948	**945**	4,970	1,143	44,145	1,099	271,073	1,066	8,000	1,018	32,785	978	358,991	953
CMT11Y	65	3,288	852	39,362	847	323,131	808	5,397	959	52,186	941	605,670	911	4,053	807	46,710	758	610,472	735
CMT12X	82	3,841	922	19,038	896	530,679	878	5,856	1,045	46,598	978	350,033	947	6,160	855	79,260	781	582,126	760
CMT12Y	69	5,334	625	43,937	570	618,451	533	5,732	771	66,070	625	479,259	597	4,428	537	38,604	495	414,698	470
CMT13X	64	4,271	859	68,074	843	611,767	825	3,043	995	17,511	949	326,223	925	2,805	850	46,648	826	684,418	809
CMT13Y	82	8,047	1,167	64,721	1,151	648,640	1,106	5,528	1,220	61,388	1,200	526,070	1,164	3,112	1,020	36,238	1,002	355,980	973
CMT14X	69	5,065	694	63,141	684	408,649	663	5,079	774	42,840	736	467,408	717	5,200	627	40,683	576	349,335	548
CMT14Y	75	3,384	604	57,643	576	876,048	555	2,786	668	31,246	649	555,779	614	5,981	523	49,552	495	190,986	501
Average		5,807		57,471		586,605		5,108		53,674		533,534		5,197		52,673		528,212	

n: number of customers, iter. : iteration of best solution based on iteration set

Table 4: Iteration of best solution of all variants

5.3 Comparison regarding efficiency

As proven in Section 5.2, the LNS algorithm, although time-consuming for the VRPDDP, yields successful results in all variants. This section delves into a comparison of efficiency. The subsequent analysis reviews how customer distribution is organised within the route plans, with empty routes being removed to focus merely on relevant routes. Table 5 compares the number of vehicles h required in the initial solution h_I with the required vehicles h_F of the improved solution after the LNS algorithm has been applied. $h\Delta$ denotes the absolute difference between both values by subtracting h_I from h_F. Due to the nature of the data, averaging is not applicable. Hence, results for each instance in Table 5 are taken exclusively from the third run. The same applies to the subsequent printed routes in Figures 10 and 11.

		VRPCB			**VRPDDP**			**VRPSPD**		
Instance	n	h_I	h_F	$h\Delta$	h_I	h_F	$h\Delta$	h_I	h_F	$h\Delta$
CMT1X	50	3	4	1	3	5	2	4	4	0
CMT1Y	50	4	5	1	3	5	2	3	4	1
CMT2X	75	8	9	1	6	8	2	7	7	0
CMT2Y	75	9	7	**-2**	6	9	3	6	7	1
CMT3X	40	2	3	1	2	3	1	2	2	0
CMT3Y	58	3	4	1	3	4	1	3	3	0
CMT4X	50	2	3	1	2	3	1	2	3	1
CMT4Y	70	4	4	0	3	5	2	4	4	0
CMT5X	73	4	5	1	3	5	2	3	4	1
CMT5Y	85	6	6	0	4	5	1	4	4	0
CMT6X	50	3	4	1	3	5	2	4	4	0
CMT6Y	50	4	5	1	3	5	2	3	4	1
CMT7X	75	8	8	0	6	9	3	7	9	2
CMT7Y	75	9	8	**-1**	6	9	3	6	8	2
CMT8X	44	2	3	1	2	3	1	2	2	0
CMT8Y	63	4	5	1	3	4	1	3	4	1
CMT9X	54	3	4	1	3	4	1	3	3	0
CMT9Y	61	3	4	1	3	4	1	3	3	0
CMT10X	72	4	5	1	3	5	2	3	5	2
CMT10Y	74	4	4	0	3	5	2	4	4	0
CMT11X	71	3	3	0	3	4	1	3	3	0
CMT11Y	65	2	3	1	3	3	0	3	2	**-1**
CMT12X	82	6	6	0	5	6	1	5	6	1
CMT12Y	69	4	3	**-1**	3	4	1	3	3	0
CMT13X	64	3	3	0	3	3	0	3	3	0
CMT13Y	82	4	5	1	3	4	1	3	3	0
CMT14X	69	5	4	**-1**	4	5	1	5	4	**-1**
CMT14Y	75	4	3	**-1**	3	4	1	3	3	0

Table 5: Comparison of routes per instance (run 3 only)

The number of routes corresponds to the number of vehicles necessary in the route plan. In some cases, the same number of vehicles is maintained (0), or even an increase in the fleet is required, as indicated by positive numbers. Only a few instances achieved a reduction in the number of vehicles, which are marked in bold. Among the variants, the VRPCB is most effective at decreasing the number of routes, while the VRPDDP tends to increase its fleet size. It is important to note that no explicit limits on the number of vehicles were implemented in the code, as this was not a specific focus of the study. Concentrating on the nodes within the printed routes of run three, which are detailed in the appendix, the routes are generally well-balanced in terms of length. Particularly, the VRPDDP demonstrates evenly divided routes in each optimized route plan. In contrast, the VRPCB shows less efficiency, with 5 out of 28 instance set solutions displaying very short routes consisting of one or two customers. Specifically, in instance CMT5Y, one route includes only node 27, likely because the customer's significant delivery demand might have exceeded the capacity constraint of most routes. This increases the total transportation costs, as one single vehicle must travel the route without including additional customer nodes, resulting in greater inefficiency. The VRPSPD, in comparison, tends to have shorter routes due to the requirement that customers have to be served simultaneously. This variant often results in a higher number of final routes than expected, despite having fewer nodes than both other variants, yet it still achieves lower overall costs.

Comparison of routes regarding paired customers

The routes in Figure 10 represent the final route solutions for instance CMT3X, which involves 40 customers in the VRPSPD without splitting and 74 customers when nodes are split for delivery and pickup demand in the VRPCB and VRPDDP variants. Locations '0' and '41' serve as the single depot for all routes.

VRPCB Final Solution Cost: 743
Route 1: 0 27 1 31 7 18 8 19 11 10 32 30 20 9 35 34 33 3 29 24 25 4 39 23 22 21 40 2 **15-56** 43 61 62 63 74 45 65 64 69 44 73 50 60 70 72 51 52 59 36 48 71 42 67 41
Route 2: 0 13 6 5 16 37 38 **14-55** 57 17 49 58 46 47 54 41
Route 3: 0 28 12 **26-66** 53 68 41

VRPDDP Final Solution Cost: 705
Route 1: 0 28 **27-67** 42 53 29 44 69 3 1 31 7 11 **59-19** 36 72 70 71 48 **51-10** 52 32 60 30 20 35 **34-73** **50-9** 33 12 68 41
Route 2: 0 **21-61** **2-43** **15-56** **63-23** **22-62** **74-65** 25 39 **4-45** **64-24** **66-26** 41
Route 3: 0 **13-54** **6-47** **18-58** 5 **8-49** 46 17 16 38 **55-14** 57 37 **75-40** 41

VRPSPD Final Solution Cost: 569
Route 1: 0 27 31 11 19 7 36 8 18 38 14 37 16 17 5 6 13 22 15 40 2 41
Route 2: 0 21 23 4 39 25 24 3 29 33 35 34 9 20 30 1 10 32 12 26 28 41

Figure 10: Final solution of instance CMT3X with 40 customers and 5 insertion options (run 3 only)

In the VRPCB setup, all delivery customers, identified by numbers below 41, are visited prior to their corresponding pickup customers, indicating that the specific VRPCB constraint is satisfied. Additionally, it is noted that customers 17 and 36 serve solely as pickup customers, as they do not require delivery services.

Regarding the VRPDDP variant, it is worth analysing whether customer pairs are visited sequentially after running the LNS algorithm or not. To address this question, such paired customer nodes are seen in highlighted numbers in Figure 10. The destroy and repair methods were reviewed by including output notes that provided insights about the ID of the particular customers that were removed and repaired. The destruction parameter was adjusted from a random selection to a fixed value of 65.00% to minimize randomness in the comparison of results. The results in Figure 10 show that 20 out of 34 (59.00%) customer pairs were visited successively. Since both customers share the exact same coordinates, visiting them consecutively makes sense as it results in zero travel units, thereby optimizing overall travel costs. This strategic routing minimizes unnecessary travel and enhances the efficiency of the route plan. The separate visits of the remaining customer pairs can be explained by the initial solution construction, where paired customers are not necessarily positioned next to each other in the route plan. Instead, they are placed according to the lowest cost, prioritizing cost reduction over keeping paired customers together. When the LNS algorithm is applied, not all paired customers are removed and reinserted due to the destroy ratio. Consequently, some customers, including their paired node, might not be repositioned during the LNS process and, therefore, maintain at separate positions until the best solution is found.

Adjustment of parameter Reinsertion Option

Overall, the LNS algorithm for the VRPDDP achieved successful results concerning the reinsertion of paired customers. Still, further optimization could be generated by fine-tuning another parameter. The Reinsertion Option parameter obtains a neighbouring reinsertion of all destroyed customer pairs. Accordingly, studying the parameter and its role in the process is essential. Before showing the adjusted results, it is crucial to revisit the implementation regarding the Reinsertion Option parameter made in the reinsertion method, as introduced in Section 4.2. After reinsertion costs are calculated during the repair method, five optimal reinsertion positions are identified, from which one is chosen at random. Consequently, the position next to the paired customer is not always selected, potentially leading to paired customers not being positioned next to each other after the LNS process. This could be another cause for the separation of some customer pairs in addition to the previously mentioned factors. An adjustment of this parameter was made to address this issue. The algorithm was subsequently run with five options, consistently selecting the least costly, hence best position at each time. This tuning aims to determine if this results in more paired customers being positioned next to each other and whether this change impacts the overall solution quality. Figure 11 depicts all three variants, with particular attention paid to the VRPDDP.

VRPCB Final Solution Cost: 733
Route 1: 0 27 1 31 10 32 30 20 9 35 34 33 3 29 **24-64** 69 44 73 50 60 70 72 51 71 42 41
Route 2: 0 28 12 26 40 21 4 25 39 23 22 2 15 14 38 16 37 5 18 8 7 19 **11-52** 59 36 48 49 58 46 17 57 55 56 43 62 63 74 65 45 61 66 53 68 67 41
Route 3: 0 6 **13-54** 47 41

VRPDDP Final Solution Cost: 705
Route 1: 0 **13-54** 40 21 63 45 **65-25 39-74 62-22** 23 4 61 27 41
Route 2: 0 67 **68-28 71-31** 7 59 11 72 **60-20 30-70** 32 19 36 52 **10-51 42-1** 33 **73-34 29-69 64-24 44-3** 50 35 9 **12-53** 66 **2-43** 75 26 41
Route 3: 0 **58-18** 46 **49-8** 48 17 **16-57** 5 **47-6** 37 38 **14-55 56-15** 41

VRPSPD Final Solution Cost: 533
Route 1: 0 2 22 4 39 25 23 13 15 38 14 37 16 17 8 5 18 6 41
Route 2: 0 27 41
Route 3: 0 1 31 10 7 19 36 11 32 20 30 9 35 34 33 3 29 24 12 21 40 28 26 41

Figure 11: Final solution of instance CMT3X with 40 customers and best insertion chosen (run 3 only)

Regarding the VRPCB, the bold text again indicates the change from delivery to pickup and justifies the correctness of the reinsertion in reference to its constraint. In the VRPDDP variant, by applying the adjustment to the parameter, all 22 customer pairs that were previously destroyed are now reinserted next to each other. Given the fixed value of the destruction parameter, this means that by tuning the parameter, all 65.00% removed customers remain as paired nodes after the reinsertion.

Five Reinsertion Options	Best Option is chosen
VRPCB: 743 VRPDDP: 705 VRPSPD: 569	VRPCB: 733 VRPDDP: 705 VRPSPD: 533

Table 6: Final Solution costs in units

Comparing the costs of both Reinsertion Options, one can see that final solution costs could be improved for VRPCB and VRPSPD, whereas the VRPDDP stays the same (Table 6). In general, only taking the best costs without having randomisation in the parameter gives better results concerning the final solution costs. The reduction is much greater in the VRPSPD variant than in the VRPDDP. This indicates once more that the LNS algorithm has successfully minimized overall transportation costs by selecting the lowest possible cost for each position.

5.4 Comparison of Destroy Operators

For the three destroy operators, performance could vary substantially. To investigate this, the code was run separately for each destroy operator. When running the code with the instances from Section 5.1 on the Smallest Demand Removal as well as on the Largest Demand Removal solely, the initial solution I_{DDP} could not be improved for the VRPDDP in most instances. In contrast, whenever the number of customers was further set below to a maximum of 50 customers per instance, the initial solution could be improved, and lower costs could be achieved. The reason for that could be that the solution space will become significantly larger with more customers. For 70 customers, this means the algorithm deals with 140 nodes in the VRPDDP; the complexity significantly increases compared to other variants where each customer is a single node. Therefore, the algorithm may have struggled to find the near-optimal routes as it has to evaluate a much larger number of possible combinations, resulting in an increase in computational complexity. So once again, the instances had to be adjusted, and the number of customers was reduced to at most 50 customers instead of the previously set limitation of 85 customers. Again, each destroy operator was run five times and then averaged.

VRP with Clustered Backhauls (VRPCB)

As before, I_{CB} shows the baseline costs before applying the LNS algorithm, whereas the final solution costs (F_{CB}) indicate the costs after applying the LNS algorithm with the respective destroy operator (Table 7). Lower values suggest a more effective optimization. All improvements of over 40.00% were achieved across all destroy operators in the instances CMT2Y, CMT7Y, CMT11X, CMT12X, CMT12Y and CMT14X. Notably, the Destroy Largest and Destroy Smallest operators reached identical improvements of 34.28%, suggesting that the specific criterion for selecting which customers to destroy (smallest demand versus largest demand) does not significantly affect the outcome. The Random destroy operator performs slightly better, indicating its robustness across different instances.

			Destroy Random		Destroy Largest		Destroy Smallest	
Instance	n	I_{CB}	F_{CB}	Δ	F_{CB}	Δ	F_{CB}	Δ
CMT1X	50	1142	828	27.46%	820	28.21%	816	28.56%
CMT1Y	50	1310	825	36.99%	819	37.47%	820	37.37%
CMT2X	49	1284	804	37.37%	791	38.43%	813	36.67%
CMT2Y	48	1393	771	44.67%	760	45.43%	752	46.04%
CMT3X	40	1071	745	30.44%	738	31.06%	749	30.08%
CMT3Y	54	1405	882	37.22%	893	36.41%	900	35.96%
CMT4X	48	1159	807	30.35%	809	30.18%	807	30.41%
CMT4Y	30	900	596	33.82%	586	34.84%	590	34.47%
CMT5X	37	1035	697	32.70%	697	32.68%	702	32.19%
CMT5Y	35	1001	651	34.95%	665	33.55%	669	33.19%
CMT6X	40	946	658	30.42%	684	27.74%	686	27.48%
CMT6Y	41	1138	685	39.77%	688	39.53%	688	39.51%
CMT7X	35	887	659	25.68%	655	26.18%	658	25.86%
CMT7Y	45	1390	746	46.33%	742	46.63%	734	47.22%
CMT8X	55	1287	890	30.86%	913	29.06%	904	29.73%
CMT8Y	38	1017	721	29.09%	714	29.77%	718	29.42%
CMT9X	51	1251	840	32.82%	859	31.32%	852	31.86%
CMT9Y	37	1007	694	31.06%	696	30.86%	697	30.78%
CMT10X	39	1035	734	29.12%	730	29.47%	740	28.46%
CMT10Y	41	1001	745	25.61%	752	24.90%	740	26.05%
CMT11X	31	726	404	44.41%	412	43.20%	412	43.31%
CMT11Y	55	1050	753	28.25%	835	20.44%	775	26.15%
CMT12X	47	819	462	43.61%	479	41.49%	469	42.74%
CMT12Y	59	1127	623	44.74%	606	46.26%	634	43.71%
CMT13X	44	933	600	35.67%	602	35.52%	582	37.60%
CMT13Y	42	765	545	28.71%	597	21.99%	563	26.46%
CMT14X	50	856	480	43.90%	486	43.22%	485	43.29%
CMT14Y	45	675	448	33.57%	441	34.67%	498	26.25%
Average		1058	689	34.84%	695	34.28%	695	34.28%

n: number of customers, I_{CB} : initial solution cost, F_{CB}: final solution cost, Δ: (I-F)/I improvement in %

Table 7: Comparison of Destroy operators for VRPCB

VRP with Divisible Pickup and Delivery (VRPDDP)

Turning to the VRP with Divisible Delivery and Pickup (VRPDDP), Table 8 presents a detailed analysis using the same three destroy operators: Random, Largest Demand, and Smallest Demand. Examining the table, the Random destroy and Destroy Smallest operators often show improvements, yet there are several instances where the LNS algorithm fails to improve the initial solution, as indicated by bold values in the Destroy Largest column. The performance of the Largest Demand Destroy could be more consistent. This inconsistency could be due to the high-demand nodes, including their paired customer, which might complicate reinsertion and, therefore, make cost optimization particularly challenging. The average improvements across all instances for each destroy operator reveal that the Random operator provides the highest average improvement (28.99%), followed by the Smallest Demand operator (25.65%), with the Largest Demand operator being the least influential (16.59%).

			Destroy Random		**Destroy Largest**		**Destroy Smallest**	
Instance	n	I_{DDP}	F_{DDP}	Δ	F_{DDP}	Δ	F_{DDP}	Δ
CMT1X	50	1226	844	31.16%	917	25.17%	946	22.81%
CMT1Y	50	1193	817	31.52%	951	20.28%	887	25.68%
CMT2X	49	1413	860	39.15%	977	30.84%	838	40.69%
CMT2Y	48	1247	807	35.30%	872	30.04%	823	33.97%
CMT3X	40	844	706	16.37%	**844**	**0.00%**	683	19.08%
CMT3Y	54	1200	931	22.43%	1054	12.18%	968	19.35%
CMT4X	48	1205	830	31.12%	961	20.23%	870	27.77%
CMT4Y	30	688	506	26.42%	627	8.90%	513	25.41%
CMT5X	37	802	655	18.30%	771	3.84%	752	6.18%
CMT5Y	35	754	629	16.63%	**754**	**0.00%**	681	9.73%
CMT6X	40	987	696	29.52%	758	23.20%	707	28.39%
CMT6Y	41	987	693	29.75%	745	24.54%	821	16.78%
CMT7X	35	1012	635	37.29%	667	34.07%	615	39.21%
CMT7Y	45	1234	770	37.63%	888	28.02%	765	38.01%
CMT8X	55	1238	930	24.89%	**1238**	**0.00%**	1006	18.72%
CMT8Y	38	862	677	21.46%	851	1.32%	719	16.64%
CMT9X	51	1209	870	28.01%	**1209**	**0.00%**	884	26.87%
CMT9Y	37	824	652	20.90%	780	5.39%	677	17.79%
CMT10X	39	840	693	17.50%	**840**	**0.00%**	673	19.93%
CMT10Y	41	968	735	24.05%	848	12.38%	767	20.72%
CMT11X	31	552	401	27.43%	410	25.72%	398	27.93%
CMT11Y	55	1088	783	28.03%	957	12.08%	784	27.90%
CMT12X	47	761	462	39.32%	446	41.42%	519	31.80%
CMT12Y	59	1037	629	39.31%	**1037**	**0.00%**	663	36.05%
CMT13X	44	895	634	29.21%	611	31.73%	686	23.35%
CMT13Y	42	651	493	24.27%	**651**	**0.00%**	511	21.44%
CMT14X	50	757	500	33.98%	502	33.71%	557	26.47%
CMT14Y	45	680	446	34.41%	483	29.03%	474	30.32%
Average		970	689	28.99%	809	16.59%	721	25.65%

n: number of customers, I_{DDP} : initial solution cost, F_{DDP}: final solution cost, Δ: (I-F)/I improvement in %

Table 8: Comparison of Destroy operators for VRPDDP

VRP with Simultaneous Pickup and Delivery (VRPSPD)

Lastly, the Destroy Operators for the VRP with Simultaneous Pickup and Delivery (VRPSPD) are analysed in Table 9. Both the Random Destroy operator as well as the Smallest Demand Destroy consistently show improvements across the instances. However, the Largest Demand Destroy operator occasionally struggles to improve the initial solution.

			Destroy Random		Destroy Largest		Destroy Smallest	
Instance	n	I_{SPD}	F_{SPD}	Δ	F_{SPD}	Δ	F_{SPD}	Δ
CMT1X	50	836	647	22.58%	674	19.33%	625	25.24%
CMT1Y	50	792	629	20.61%	646	18.41%	605	23.61%
CMT2X	49	754	649	13.93%	**754**	**0.00%**	652	13.58%
CMT2Y	48	798	607	23.91%	635	20.48%	614	23.08%
CMT3X	40	647	526	18.67%	584	9.71%	555	14.19%
CMT3Y	54	860	700	18.58%	668	22.33%	658	23.44%
CMT4X	48	768	632	17.71%	653	14.92%	630	18.02%
CMT4Y	30	496	393	20.81%	388	21.77%	378	23.75%
CMT5X	37	576	527	8.54%	569	1.22%	550	4.58%
CMT5Y	35	545	462	15.30%	456	16.29%	451	17.32%
CMT6X	40	662	537	18.94%	572	13.53%	546	17.46%
CMT6Y	41	683	529	22.58%	549	19.56%	519	24.04%
CMT7X	35	654	484	25.93%	522	20.18%	490	25.11%
CMT7Y	45	801	574	28.29%	605	24.42%	578	27.89%
CMT8X	55	903	696	22.88%	710	21.37%	693	23.21%
CMT8Y	38	601	529	12.01%	555	7.65%	530	11.75%
CMT9X	51	858	672	21.66%	699	18.53%	659	23.15%
CMT9Y	37	608	510	16.05%	512	15.86%	521	14.31%
CMT10X	39	643	522	18.76%	587	8.65%	554	13.78%
CMT10Y	41	695	538	22.53%	574	17.44%	564	18.85%
CMT11X	31	517	357	30.87%	362	30.06%	354	31.61%
CMT11Y	55	904	630	30.33%	637	29.56%	614	32.12%
CMT12X	47	461	368	20.26%	388	15.75%	337	26.81%
CMT12Y	59	631	496	21.39%	559	11.44%	510	19.21%
CMT13X	44	673	494	26.57%	475	29.45%	471	30.07%
CMT13Y	42	504	437	13.25%	449	10.83%	433	14.01%
CMT14X	50	506	387	23.52%	406	19.80%	348	31.26%
CMT14Y	45	545	349	35.96%	366	32.92%	353	35.19%
Average		676	532	21.34%	556	17.79%	528	21.82%

n: number of customers, I_{SPD} : initial solution cost, F_{SPD}: final solution cost, Δ: (I-F)/I improvement in %

Table 9: Comparison of Destroy operators for VRPSPD

This is specifically notable in instance CMT2X, though this occurrence is less frequent compared to the VRPDDP. This time, the Destroy Smallest demand operator provide the highest average improvements across all instances. Overall, the analysis of the three destroy operators—Random, Largest Demand, and Smallest Demand—across the VRPCB, VRPDDP, and VRPSPD variants reveals distinct patterns in their effectiveness. All operators achieve positive average improvements. However, the Largest Demand destroy operator demonstrates inconsistent results, often being the least effective, particularly in the VRPDDP variant. While no specific operator stands out significantly, each operator achieves good results across different instances.

6 Conclusion

This paper has introduced the Vehicle Routing Problem with Backhauls (VRPB), a subclass of the Capacitated VRP, with a focus on three variants: the VRP with Clustered Backhauls (VRPCB), the VRP with Divisible Pickup and Delivery (VRPDDP) and the VRP with Simultaneous Pickup and Delivery (VRPSPD). The primary objective was to compare these variants regarding the reduction of overall transportation costs within the route plan of each instance. To achieve this, the Large Neighborhood Search (LNS) algorithm, a local search-based metaheuristic, was employed to optimize the initial solutions. The initial route plans were generated using the Sequential Insertion Heuristic combined with a Forward Labelling algorithm to ensure capacity and feasibility constraints were met. The implemented LNS algorithm includes three predefined destroy operators inside the destroy method, namely the Random Removal, Largest Demand and Smallest Demand removal, followed by the repair method to reinsert the removed nodes. After running the code for 1,000,000 iterations, the results demonstrate that the LNS algorithm successfully improved the solutions, resulting in final costs that were consistently reduced compared to the respective initial solution costs across all instances. The VRPCB variant achieved the most significant improvements (in %) when comparing the initial to the final solution costs. In contrast, the VRPSPD variant produced the lowest overall final solution costs relative to the final solution costs of the other variants. Although the VRPDDP effectively split customers into two nodes, it faced challenges in reducing costs to the same extent or even more as both other variants. Adjusting the Reinsertion Options parameter led to more paired customers being positioned next to each other, contributing to further reduction in costs. However, the cost reduction of the VRPDDP could not surpass the performance of both other variants, as was initially expected. In terms of the destroy method, all three destroy operators yielded positive improvements, with no single operator proving to be significantly superior to the others. In conclusion, the LNS algorithm efficiently reduced transportation costs, thereby fulfilling the objective function of this thesis. The relevance of the problems and variants explored in this research have been growing in recent years, as the combination of pickup and delivery operations has become essential for companies aiming to reduce costs. This subject will remain significant as reverse logistics continues to be an important field of study.

7 References

Alfredo Tang Montané, F., & Galvão, R. D. (2006). A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Computers & Operations Research*, *33*(3), 595–619. https://doi.org/10.1016/j.cor.2004.07.009

Avci, M., & Topaloglu, S. (2015). An adaptive local search algorithm for vehicle routing problem with simultaneous and mixed pickups and deliveries. *Computers & Industrial Engineering*, *83*, 15–29. https://doi.org/10.1016/j.cie.2015.02.002

Battarra, M., Cordeau, J.-F., & Iori, M. (2014). *Chapter 6: Pickup-and-Delivery Problems for Goods Transportation* (P. Toth & D. Vigo, Eds.; 161–191. Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9781611973594.ch6

Belloso, J., Juan, A. A., Martinez, E., & Faulin, J. (2017). A biased-randomized metaheuristic for the vehicle routing problem with clustered and mixed backhauls. *Networks*, *69*(3), 241–255. https://doi.org/10.1002/net.21734

Bianchessi, N., & Righini, G. (2007). Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery. *Computers & Operations Research*, *34*(2), 578–594. https://doi.org/10.1016/j.cor.2005.03.014

Braekers, K., Ramaekers, K., & Van Nieuwenhuyse, I. (2016). The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*, *99*, 300–313. https://doi.org/10.1016/j.cie.2015.12.007

Bräysy, O., & Gendreau, M. (2005). Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms. *Transportation Science*, *39*(1), 104–118.

Cabrera-Guerrero, P., Moltedo-Perfetti, A., Cabrera, E., & Paredes, F. (2016). Comparing Two Heuristic Local Search Algorithms for a Complex Routing Problem. *Studies in Informatics and Control*, *25*(4). https://doi.org/10.24846/v25i4y201602

Chentli, H., Ouafi, R., & Cherif-Khettaf, W. R. (2018). A selective adaptive large neighborhood search heuristic for the profitable tour problem with simultaneous pickup and delivery services. *RAIRO - Operations Research*, *52*(4–5), 1295–1328. https://doi.org/10.1051/ro/2018024

Cordeau, J.-F., Gendreau, M., Laporte, G., Potvin, J.-Y., & Semet, F. (2002). A guide to vehicle routing heuristics. *Journal of the Operational Research Society*, *53*(5), 512–522. https://doi.org/10.1057/palgrave.jors.2601319

Crainic, T. G., & Toulouse, M. (2010). Parallel Meta-heuristics. In M. Gendreau & J.-Y. Potvin (Eds.), *Handbook of Metaheuristics,* 497–541. Springer US. https://doi.org/10.1007/978-1-4419-1665-5_17

Demirbilek, M. (2021). A metaheuristic solution approach with two construction heuristics for vehicle routing problem with simultaneous Linehauls and Backhauls. *Sigma Journal of Engineering and Natural Sciences – Sigma Mühendislik ve Fen Bilimleri Dergisi*. https://doi.org/10.14744/sigma.2021.00012

Desaulniers, G., Errico, F., Irnich, S., & Schneider, M. (2016). Exact Algorithms for Electric Vehicle-Routing Problems with Time Windows. *Operations Research*, *64*. https://doi.org/10.1287/opre.2016.1535

Dethloff, J. (2001). Vehicle routing and reverse logistics: The vehicle routing problem with simultaneous delivery and pick-up. *OR-Spektrum*, *23*(1), 79–96. https://doi.org/10.1007/PL00013346

Dinc Yalcın, G., & Erginel, N. (2015). Fuzzy multi-objective programming algorithm for vehicle routing problems with backhauls. *Expert Systems with Applications*, *42*(13), 5632–5644. https://doi.org/10.1016/j.eswa.2015.02.060

Garey, M. R. (with Internet Archive). (1979). *Computers and intractability: A guide to the theory of NP-completeness*. San Francisco : W. H. Freeman. http://archive.org/details/computersintract0000gare

Gendreau, M., & Potvin, J.-Y. (2010). Tabu Search. In M. Gendreau & J.-Y. Potvin (Eds.), *Handbook of Metaheuristics*, 41–59. Springer US. https://doi.org/10.1007/978-1-4419-1665-5_2

Granada-Echeverri, M., Cubides, L., & Bustamante, O. (2019). The electric vehicle routing problem with backuals. *International Journal of Industrial Engineering Computations*, *10*, 1–22. https://doi.org/10.5267/j.ijiec.2019.6.001

Haddad, M. N., Martinelli, R., Vidal, T., Ochi, L. S., Martins, S., Souza, M. J. F., & Hartl, R. (2018). *Large Neighborhood-Based Metaheuristic and Branch-and-Price for the Pickup and Delivery Problem with Split Loads* (arXiv:1802.06318). arXiv. http://arxiv.org/abs/1802.06318

Koç, Ç., & Laporte, G. (2018). Vehicle routing with backhauls: Review and research perspectives. *Computers & Operations Research*, *91*, 79–91. https://doi.org/10.1016/j.cor.2017.11.003

Konstantakopoulos, G. D., Gayialis, S. P., & Kechagias, E. P. (2022). Vehicle routing problem and related algorithms for logistics distribution: A literature review and classification. *Operational Research*, *22*(3), 2033–2062. https://doi.org/10.1007/s12351-020-00600-7

Laporte, G., Ropke, S., & Vidal, T. (2014). *Chapter 4: Heuristics for the Vehicle Routing Problem*, 87–116. https://doi.org/10.1137/1.9781611973594.ch4

Lei, H., Laporte, G., & Guo, B. (2011). The capacitated vehicle routing problem with stochastic demands and time windows. *Computers & Operations Research*, *38*(12), 1775–1783. https://doi.org/10.1016/j.cor.2011.02.007

Lenstra, J. K., & Kan, A. H. G. R. (1981). Complexity of vehicle routing and scheduling problems. *Networks*, *11*(2), 221–227. https://doi.org/10.1002/net.3230110211

Lourenço, H. R., Martin, O. C., & Stützle, T. (2010). Iterated Local Search: Framework and Applications. In M. Gendreau & J.-Y. Potvin (Eds.), *Handbook of Metaheuristics*, 363–397. Springer US. https://doi.org/10.1007/978-1-4419-1665-5_12

Marinakis, Y., & Migdalas, A. (2007). *Annotated bibliography in vehicle routing.* https://doi.org/10.1007/BF02941184

Mekamcha, K., Souier, M., Bessenouci, H. N., & Bennekrouf, M. (2021). Two metaheuristics approaches for solving the traveling salesman problem: An Algerian waste collection case. *Operational Research*, *21*(3), 1641–1661. https://doi.org/10.1007/s12351-019-00529-6

Min, H. (1989). The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transportation Research Part A: General*, *23*(5), 377–386. https://doi.org/10.1016/0191-2607(89)90085-X

Mingozzi, A., Giorgi, S., & Baldacci, R. (1999). An Exact Method for the Vehicle Routing Problem with Backhauls. *Transportation Science*, *33*, 315–329. https://doi.org/10.1287/trsc.33.3.315

Mitrović-Minić, S., & Laporte, G. (2004). Waiting strategies for the dynamic pickup and delivery problem with time windows. *Transportation Research Part B: Methodological*, *38*(7), 635–655. https://doi.org/10.1016/j.trb.2003.09.002

Mosheiov, G. (1998). Vehicle routing with pick-up and delivery: Tour-partitioning heuristics. *Computers & Industrial Engineering*, *34*(3), 669–684. https://doi.org/10.1016/S0360-8352(97)00275-1

Nagy, G., & Salhi, S. (2005). Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *European Journal of Operational Research*, *162*(1), 126–141. https://doi.org/10.1016/j.ejor.2002.11.003

Nagy, G., Wassan, N. A., Speranza, M. G., & Archetti, C. (2015). The Vehicle Routing Problem with Divisible Deliveries and Pickups. *Transportation Science*, *49*(2), 271–294. https://doi.org/10.1287/trsc.2013.0501

Parragh, S. N., Doerner, K. F., & Hartl, R. F. (2008). A survey on pickup and delivery problems. *Journal Für Betriebswirtschaft*, *58*(1), 21–51. https://doi.org/10.1007/s11301-008-0033-7

Pisinger, D., & Ropke, S. (2010). Large Neighborhood Search. In M. Gendreau & J.-Y. Potvin (Eds.), *Handbook of Metaheuristics*, 399–419. Springer US. https://doi.org/10.1007/978-1-4419-1665-5_13

Potvin, J.-Y., Duhamel, C., & Guertin, F. (1996). A Genetic Algorithm for Vehicle Routing with Backhauling. *Applied Intelligence*, *6*, 345–355. https://doi.org/10.1007/BF00132738

Resende, M. G. C., & Ribeiro, C. C. (2010). Greedy Randomized Adaptive Search Procedures: Advances, Hybridizations, and Applications. In M. Gendreau & J.-Y. Potvin (Eds.), *Handbook of Metaheuristics*, 283–319. Springer US. https://doi.org/10.1007/978-1-4419-1665-5_10

Ropke, S., & Pisinger, D. (2006). A unified heuristic for a large class of Vehicle Routing Problems with Backhauls. *European Journal of Operational Research*, *171*(3), 750–775. https://doi.org/10.1016/j.ejor.2004.09.004

Salhi, S., & Nagy, G. (1999). A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. *Journal of the Operational Research Society*, *50*(10), 1034–1042. https://doi.org/10.1057/palgrave.jors.2600808

Sar, K., & Ghadimi, P. (2023). A systematic literature review of the vehicle routing problem in reverse logistics operations. *Computers & Industrial Engineering*, *177*, 109011. https://doi.org/10.1016/j.cie.2023.109011

Savelsbergh, M. W. P., & Sol, M. (1995). The General Pickup and Delivery Problem. *Transportation Science*, *29*(1), 17–29. https://doi.org/10.1287/trsc.29.1.17

Schrimpf, G., Schneider, J., Stamm-Wilbrandt, H., & Dueck, G. (2000). Record Breaking Optimization Results Using the Ruin and Recreate Principle. *Journal of Computational Physics*, *159*(2), 139–171. https://doi.org/10.1006/jcph.1999.6413

Shaw, J., Delery, J., Jenkins, G., & Gupta, N. (1998). An Organization-Level Analysis of Voluntary and Involuntary Turnover. *Academy of Management Journal*, *41*, 511–525. https://doi.org/10.2307/256939

Shaw, P. (1998). Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems. In M. Maher & J.-F. Puget (Eds.), *Principles and Practice of Constraint Programming—CP98*, 417–431. Springer. https://doi.org/10.1007/3-540-49481-2_30

Solomon, M. M. (1987). Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. *Operations Research*, *35*(2), 254–265. https://doi.org/10.1287/opre.35.2.254

Souza, M. J. F., Mine, M. T., Silva, M. de S. A., Ochi, L. S., & Subramanian, A. (2011). A hybrid heuristic, based on Iterated Local Search and GENIUS, for the Vehicle Routing Problem with Simultaneous Pickup and Delivery. *International Journal of Logistics Systems and Management*, *10*(2), 142–157. https://doi.org/10.1504/IJLSM.2011.042625

Tan, S.-Y., & Yeh, W.-C. (2021). The Vehicle Routing Problem: State-of-the-Art Classification and Review. *Applied Sciences*, *11*(21). https://doi.org/10.3390/app112110295

Toth, P., & Vigo, D. (1997). An Exact Algorithm for the Vehicle Routing Problem with Backhauls. *Transportation Science*, *31*(4), 372–385. https://doi.org/10.1287/trsc.31.4.372

Wassan, N., & Nagy, G. (2014). Vehicle Routing Problem with Deliveries and Pickups: Modelling Issues and Meta-heuristics Solution Approaches. *International Journal of Transportation*, *2*(1), 95–110. https://doi.org/10.14257/ijt.2014.2.1.06

Zachariadis, E. E., & Kiranoudis, C. T. (2011). A local search metaheuristic algorithm for the vehicle routing problem with simultaneous pick-ups and deliveries. *Expert Systems with Applications*, *38*(3), 2717–2726. https://doi.org/10.1016/j.eswa.2010.08.061

III Appendix

Appendix A: Run three of VRPCB

Instance	Initial Solution Routes	Final Solution Routes
CMT1X	Route 1:0 36 35 3 31 43 26 28 20 29 34 39 45 33 10 30 21 2 65 67 76 60 79 64 89 87 86 62 85 90 84 80 75 74 72 57 88 77 53 81 51 Route 2:0 91 61 82 17 37 44 15 55 38 83 49 9 16 50 59 52 73 78 22 68 58 8 48 93 56 69 23 7 24 70 14 63 54 13 41 42 19 40 92 66 25 71 51 Route 3:0 4 18 47 12 5 11 32 1 27 6 46 51	Route 1: 0 46 11 32 1 22 68 52 78 73 51 Route 2: 0 12 47 18 14 25 13 41 19 40 42 4 17 37 44 15 45 33 79 90 64 82 89 87 85 86 62 71 63 66 54 92 61 91 51 Route 3: 0 5 38 9 49 10 39 30 34 50 21 29 2 16 65 75 67 80 76 84 60 94 59 83 55 51 Route 4: 0 27 6 48 7 23 24 43 26 8 31 28 3 20 35 36 81 53 74 77 72 58 93 57 69 88 70 56 51
CMT1Y	Route 1:0 40 41 42 33 10 34 36 3 31 43 7 26 28 35 29 21 50 30 39 45 19 13 18 70 62 86 90 84 76 67 75 20 74 72 77 53 81 80 60 79 87 85 51 Route 2:0 65 83 89 82 64 59 2 68 58 93 57 69 88 71 51 Route 3:0 4 54 63 14 25 24 23 48 6 56 66 92 12 61 37 17 44 15 49 16 9 38 5 55 52 1 8 22 78 11 32 73 91 46 51 Route 4:0 27 47 51	Route 1: 0 12 61 51 Route 2: 0 47 4 18 14 25 13 41 19 40 42 17 37 44 15 45 33 79 90 64 82 89 87 85 86 62 71 63 66 54 92 51 Route 3: 0 6 48 23 24 43 7 26 8 31 28 3 35 36 81 20 53 74 77 58 72 57 93 69 88 70 56 51 Route 4: 0 27 1 22 32 11 16 29 21 50 34 30 9 38 5 49 10 39 84 60 94 55 83 65 59 76 80 67 75 2 68 52 78 73 51 Route 5: 0 46 91 51
CMT2X	Route 1:0 66 31 25 50 49 24 41 69 61 23 56 18 55 65 59 11 127 112 129 92 130 97 116 117 115 98 124 99 105 139 85 133 140 76 Route 2:0 134 94 89 111 144 71 110 143 135 136 145 96 138 95 76 Route 3:0 113 132 109 93 131 121 80 87 128 88 84 83 118 78 90 137 123 106 76 Route 4:0 100 141 79 103 101 13 54 82 120 108 126 119 104 64 42 22 43 1 73 107 125 91 86 72 114 81 142 77 122 70 36 21 47 5 57 15 37 20 60 102 28 76 Route 5:0 53 10 39 32 44 9 58 38 14 76 Route 6:0 19 27 48 62 74 29 52 8 7 35 67 3 76 Route 7:0 63 16 40 12 51 33 2 68 30 45 76 Route 8:0 46 34 4 75 6 17 26 76	Route 1: 0 54 13 57 15 5 29 48 30 45 34 46 52 27 101 119 103 80 121 95 122 104 146 102 135 96 136 145 107 81 76 Route 2: 0 31 10 38 65 11 66 140 85 139 112 84 132 105 113 83 118 78 106 114 86 76 Route 3: 0 17 40 12 58 72 39 9 32 44 3 49 24 18 50 25 55 129 99 92 124 76 Route 4: 0 67 141 76 Route 5: 0 75 68 6 2 74 28 61 22 62 21 47 36 37 20 70 60 69 143 110 71 134 144 94 131 89 111 76 Route 6: 0 26 100 76 Route 7: 0 4 77 142 79 76 Route 8: 0 51 16 63 33 73 1 43 23 56 41 42 64 138 116 115 117 97 130 98 123 137 90 125 91 76 Route 9: 0 35 19 8 7 53 14 59 133 88 127 109 93 128 87 82 126 108 120 76
CMT2Y	Route 1:0 66 31 25 50 49 42 23 24 18 55 65 59 14 11 127 112 129 92 130 138 116 117 97 115 98 123 124 99 105 139 85 133 140 76 Route 2:0 134 143 135 96 145 95 121 110 71 111 131 89 94 144 128 93 109 88 72 76 Route 3:0 80 102 137 78 106 118 90 107 136 122 103 119 101 87 79 76 Route 4:0 132 83 125 114 113 84 120 108 126 82 76 Route 5:0 104 60 20 15 37 70 36 47 69 61 22 64 56 43 41 77 142 81 86 91 100 8 141 76 Route 6:0 54 19 13 5 28 1 63 16 73 62 74 21 57 67 76 Route 7:0 44 9 10 58 35 53 38 39 32 3 33 2 68 6 76 Route 8:0 48 30 45 52 34 7 46 27 29 4 75 51 40 12 26 76 Route 9:0 17 76	Route 1: 0 26 67 141 100 76 Route 2: 0 4 45 29 15 57 5 48 47 21 69 36 37 20 70 60 71 134 144 111 94 110 143 95 121 76 Route 3: 0 2 30 74 28 61 22 62 73 1 43 42 41 64 138 116 115 117 96 135 102 136 145 137 107 77 146 104 79 76 Route 4: 0 31 10 58 38 65 11 66 59 133 88 127 85 140 139 112 84 72 132 76 Route 5: 0 17 12 40 3 44 9 39 32 50 25 55 18 92 124 129 99 105 113 83 106 118 78 114 86 91 76 Route 6: 0 75 68 6 51 33 63 16 49 24 56 23 130 97 98 123 90 125 81 142 76 Route 7: 0 53 14 19 35 7 8 46 34 52 27 13 54 87 128 131 89 80 122 103 119 101 126 108 120 82 109 93 76
CMT3X	Route 1:0 38 16 14 15 22 39 4 24 34 3 33 9 32 30 10 7 8 19 11 20 35 29 25 23 13 47 57 36 52 60 73 64 63 56 62 43 61 45 65 69 44 50 72 70 42 71 51 59 48 58 49 46 17 55 41 Route 2:0 37 5 18 6 54 2 21 40 66 26 12 53 28 68 1 31 67 27 41	Route 1: 0 27 1 31 7 18 8 19 11 10 32 30 20 9 35 34 33 3 29 24 25 4 39 23 22 21 40 2 15 56 43 61 62 63 74 45 65 64 69 44 73 50 60 70 72 51 52 59 36 48 71 42 67 41 Route 2: 0 13 6 5 16 37 38 14 55 57 17 49 58 46 47 54 41 Route 3: 0 28 12 26 66 53 68 41
CMT3Y	Route 1:0 49 36 46 8 45 47 11 20 30 51 33 9 29 25 23 41 43 44 16 14 42 15 22 56 4 39 55 24 34 35 32 19 48 98 38 73 74 96 97 17 67 100 70 78 88 102 91 68 90 77 99 101 59 Route 2:0 92 63 71 105 106 87 50 60 89 103 76 66 69 62 82 83 107 94 75 64 95 61 65 72 104 93 79 80 81 59 Route 3:0 53 26 58 40 84 57 37 13 6 5 2 21 54 12 3 86 28 1 10 31 7 18 52 85 27 59	Route 1: 0 52 103 76 85 59 Route 2: 0 27 28 26 12 71 86 59 Route 3: 0 53 6 13 2 57 15 43 42 37 14 44 16 5 45 8 18 7 48 47 46 36 49 19 11 70 77 101 99 66 100 67 98 17 64 75 97 38 73 95 96 74 108 61 72 65 59 Route 4: 0 58 40 21 22 41 23 56 39 4 25 55 54 24 29 3 33 34 35 9 51 20 30 32 10 31 1 60 50 89 69 90 88 78 102 68 91 62 87 82 105 106 83 92 63 107 81 80 94 79 84 93 104 59
CMT4X	Route 1:0 38 16 44 43 15 23 25 29 35 20 49 47 8 45 48 7 19 11 10 32 9 34 24 4 39 22 41 42 14 13 57 65 67 90 36 69 61 82 60 54 79 75 73 84 74 83 70 62 91 46 59 17 89 66 86 87 88 51 Route 2:0 6 56 37 5 68 18 58 81 31 52 1 50 3 33 30 80 63 12 55 21 71 26 76 77 27 28 78 85 40 64 2 53 72 51	Route 1: 0 13 6 18 5 45 8 48 47 49 11 19 7 31 10 32 30 20 9 35 34 33 3 29 24 74 79 54 83 60 70 82 80 52 81 61 62 69 58 92 91 36 46 59 90 17 56 68 57 64 51 Route 2: 0 1 50 12 26 40 2 21 4 25 39 23 22 41 15 43 42 37 14 38 44 16 67 89 65 87 88 66 86 72 73 84 75 55 71 53 85 76 63 78 51 Route 3: 0 28 27 77 51

CMT4Y	Route 1:0 49 47 45 8 46 36 64 63 65 34 24 55 25 39 41 15 23 67 29 51 33 9 35 66 20 30 32 11 10 62 19 48 110 38 87 17 79 112 111 89 82 102 100 90 103 118 80 114 129 128 126 127 113 71 Route 2:0 130 95 117 131 74 99 94 104 75 91 119 106 108 109 123 124 85 107 73 120 86 92 93 71 Route 3:0 125 7 78 76 60 5 61 16 44 43 57 56 4 22 42 14 37 59 122 84 6 77 88 18 115 101 31 70 81 50 1 72 98 28 97 116 40 105 21 2 121 96 54 68 3 12 83 71 Route 4:0 27 52 69 26 53 58 13 71	Route 1: 0 52 18 7 48 19 49 36 47 46 8 45 60 5 61 16 44 14 42 57 15 43 86 107 108 85 38 109 87 124 122 84 77 123 76 17 110 79 88 115 71 Route 2: 0 28 27 69 1 30 70 31 10 62 11 64 63 32 20 66 9 51 33 34 35 65 128 129 90 100 102 81 125 126 82 127 113 89 111 112 78 101 72 50 132 97 71 Route 3: 0 53 58 40 21 26 12 68 3 29 24 54 4 55 25 39 67 23 56 41 22 92 106 93 119 75 104 130 95 118 117 94 99 103 80 114 74 131 83 96 98 71 Route 4: 0 6 59 37 13 2 120 73 91 105 121 116 71
CMT5X	Route 1:0 38 14 42 41 22 23 56 39 55 29 65 66 71 9 35 34 24 25 67 15 43 44 16 61 59 17 46 114 92 129 131 137 106 97 121 98 102 83 117 132 93 136 103 105 85 116 130 36 113 127 90 88 110 89 111 112 74 Route 2:0 96 109 76 123 95 122 78 107 133 79 80 126 82 81 104 84 128 115 74 Route 3:0 100 101 119 124 87 37 125 5 60 91 118 49 47 8 45 48 7 19 62 10 11 32 30 70 51 20 63 64 135 75 33 3 77 120 54 4 72 73 21 94 108 99 86 68 134 74 Route 4:0 57 2 13 6 18 52 31 69 1 50 12 26 40 58 53 28 27 74	Route 1: 0 53 58 40 26 12 86 99 108 124 119 74 Route 2: 0 6 13 59 37 42 15 57 2 43 14 38 44 16 61 5 60 18 8 45 113 82 46 17 127 90 112 88 110 89 111 125 79 126 91 81 104 118 74 Route 3: 0 28 101 74 Route 4: 0 27 69 1 50 33 3 68 29 24 21 72 73 22 41 56 39 4 54 55 25 67 23 96 133 107 98 121 120 78 122 94 138 95 109 123 76 87 80 74 Route 5: 0 52 7 48 47 49 64 63 11 19 62 10 31 70 30 32 20 51 9 34 35 65 71 66 132 131 137 83 106 102 97 134 77 117 93 105 129 85 130 116 36 114 115 92 128 84 103 136 75 135 100 74
CMT5Y	Route 1:0 44 43 15 67 25 24 34 35 9 71 20 49 47 46 36 64 66 65 29 55 39 56 22 74 75 23 41 42 14 16 61 85 6 17 128 126 104 97 117 144 143 141 142 125 139 102 124 123 101 100 38 86 Route 2:0 145 110 109 118 156 155 89 78 95 129 115 148 96 105 149 114 133 132 90 119 134 150 151 152 107 121 153 108 86 Route 3:0 157 98 50 140 116 130 93 158 159 103 138 92 99 137 135 106 111 120 131 136 88 122 91 94 127 87 147 154 146 86 Route 4:0 112 63 19 7 82 8 83 60 84 45 48 11 62 10 70 30 81 79 77 80 3 33 51 32 113 86 Route 5:0 57 72 54 12 28 1 69 52 31 76 68 4 21 26 58 40 73 2 37 5 18 59 13 86 Route 6:0 53 27 86	Route 1: 0 27 86 Route 2: 0 58 40 75 41 22 74 72 73 21 4 56 39 23 67 145 119 110 133 132 90 152 121 107 153 150 106 151 134 108 86 Route 3: 0 28 26 12 68 80 29 24 25 55 54 3 77 76 79 33 81 51 9 34 35 71 66 65 143 149 144 117 115 105 129 95 78 118 109 114 89 155 156 146 157 98 154 50 87 148 116 147 112 86 Route 4: 0 13 2 57 42 15 43 14 44 16 61 85 37 59 6 60 5 84 83 18 8 46 45 125 94 159 138 92 99 137 91 160 17 139 102 124 38 100 122 123 101 135 88 120 136 111 113 86 Route 5: 0 52 31 69 1 70 30 20 32 10 62 7 82 48 47 36 19 11 63 64 49 128 142 97 141 96 140 104 126 127 93 158 103 130 86 Route 6: 0 53 131 86
CMT6X	Route 1:0 36 35 3 31 43 26 28 20 29 34 39 45 33 10 30 21 2 65 67 76 60 79 64 89 87 86 62 85 90 84 80 75 74 72 57 88 77 53 81 51 Route 2:0 91 61 82 17 37 44 15 55 38 83 49 9 16 50 59 52 73 78 22 68 58 8 48 93 56 69 23 7 24 70 14 63 54 13 41 42 19 40 92 66 25 71 51 Route 3:0 4 18 47 12 5 11 32 1 27 6 46 51	Route 1: 0 5 38 49 9 50 16 2 29 21 34 30 39 10 33 45 15 44 42 87 89 82 64 90 79 84 60 76 80 67 75 65 59 94 83 55 51 Route 2: 0 46 12 47 37 17 4 18 13 41 19 40 85 86 62 71 63 66 54 92 61 91 51 Route 3: 0 27 6 14 25 24 43 7 23 48 26 8 31 28 3 35 20 36 81 53 74 77 58 72 57 88 70 69 93 56 73 51 Route 4: 0 11 32 1 22 68 52 78 51
CMT6Y	Route 1:0 40 41 42 33 10 34 36 3 31 43 7 26 28 35 29 21 50 30 39 45 19 13 18 70 62 86 90 84 76 67 75 20 74 72 77 53 81 80 60 79 87 85 51 Route 2:0 65 83 89 82 64 59 2 68 58 93 57 69 88 71 51 Route 3:0 4 54 63 14 25 24 23 48 6 56 66 92 12 61 37 17 44 15 49 16 9 38 5 55 52 1 8 22 78 11 32 73 91 46 51 Route 4:0 27 47 51	Route 1: 0 12 47 4 18 13 41 19 40 85 86 62 71 63 66 54 92 61 51 Route 2: 0 6 14 25 24 43 7 23 48 26 8 31 28 3 35 36 81 20 53 74 77 58 72 57 88 70 69 93 56 51 Route 3: 0 46 91 51 Route 4: 0 27 1 22 68 52 78 73 51 Route 5: 0 32 11 38 9 49 5 16 29 21 50 34 30 39 10 33 45 15 44 37 17 42 87 89 82 64 90 79 60 84 76 80 67 75 2 65 59 94 55 83 51
CMT7X	Route 1:0 66 31 25 50 49 24 41 69 61 23 56 18 55 65 59 11 127 112 129 92 130 97 116 117 115 98 124 99 105 139 85 133 140 76 Route 2:0 134 94 89 111 144 71 110 143 135 136 145 96 138 9576 Route 3:0 113 132 109 93 131 121 80 87 128 88 84 83 118 78 90 137 123 106 76 Route 4:0 100 141 79 103 101 13 54 82 120 108 126 119 104 64 42 22 43 1 73 107 125 91 86 72 114 81 142 77 122 70 36 21 47 5 57 15 37 20 60 102 28 76 Route 5:0 53 10 39 32 44 9 58 38 14 76 Route 6:0 19 27 48 62 74 29 52 8 7 35 67 3 76 Route 7:0 63 16 40 12 51 33 2 68 30 45 76 Route 8:0 46 34 4 75 6 17 26 76	Route 1: 0 12 39 32 3 44 50 18 124 83 113 114 86 91 100 76 Route 2: 0 67 26 17 40 9 72 58 10 31 25 55 129 99 106 78 118 92 98 123 97 130 115 116 117 145 107 137 90 125 76 Route 3: 0 7 35 8 19 59 133 82 93 109 120 126 108 141 76 Route 4: 0 4 30 48 29 45 5 37 20 70 60 36 69 47 21 74 28 22 61 135 95 121 122 80 110 143 71 134 144 94 111 76 Route 5: 0 75 68 2 62 73 1 63 33 6 81 142 76 Route 6: 0 38 65 66 11 53 14 88 127 85 140 139 112 105 84 132 76 Route 7: 0 46 34 52 27 15 57 13 54 128 87 131 89 101 76 Route 8: 0 51 16 49 24 56 23 43 42 41 64 138 96 136 102 146 77 104 103 119 79 76

CMT7Y	Route 1:0 66 31 25 50 49 42 23 24 18 55 65 59 14 11 127 112 129 92 130 138 116 117 97 115 98 123 124 99 105 139 85 133 140 76 Route 2:0 134 143 135 96 145 95 121 110 71 111 131 89 94 144 128 93 109 88 72 76 Route 3:0 80 102 137 78 106 118 90 107 136 122 103 119 101 87 79 76 Route 4:0 132 83 125 114 113 84 120 108 126 82 76 Route 5:0 104 60 20 15 37 70 36 47 69 61 22 64 56 43 41 77 142 81 86 91 100 8 141 76 Route 6:0 54 19 13 5 28 1 63 16 73 62 74 21 57 67 76 Route 7:0 44 9 10 58 35 53 38 39 32 3 33 2 68 6 76 Route 8:0 48 30 45 52 34 7 46 27 29 4 75 51 40 12 26 76 Route 9:0 17 76	Route 1: 0 37 20 70 60 69 36 47 21 48 5 29 45 15 57 131 89 80 111 94 144 134 71 143 110 121 122 95 146 77 76 Route 2: 0 17 3 44 50 18 32 9 39 31 25 55 129 99 92 124 106 83 113 105 84 132 72 76 Route 3: 0 26 12 40 114 86 100 76 Route 4: 0 75 68 51 6 2 62 1 73 33 63 16 49 24 98 123 78 118 91 125 81 142 79 76 Route 5: 0 30 74 28 61 22 64 42 41 43 23 56 130 97 117 116 115 138 96 135 102 136 145 107 90 137 76 Route 6: 0 4 76 Route 7: 0 67 34 27 52 46 35 7 8 19 54 13 87 128 93 109 82 120 108 126 101 104 103 119 141 76 Route 9: 0 58 10 38 65 66 11 53 14 59 133 88 127 85 140 112 139 76
CMT8X	Route 1:0 38 16 44 43 15 23 25 29 35 20 11 19 10 32 9 33 3 34 24 4 39 22 41 42 14 13 47 81 61 36 56 64 77 68 78 80 60 66 67 49 69 73 48 54 76 74 46 55 63 52 62 53 17 59 45 Route 2:0 27 71 1 30 31 7 75 57 12 72 28 26 70 65 21 40 79 2 58 6 51 37 5 18 8 50 45	Route 1: 0 27 71 45 Route 2: 0 28 12 26 21 40 2 13 6 18 8 5 37 16 44 38 14 42 43 15 41 22 23 39 4 25 69 49 78 67 66 80 60 82 81 59 61 17 50 51 58 47 65 79 70 57 72 45 Route 3: 0 1 30 10 31 7 19 11 32 20 9 35 34 33 3 29 24 68 73 48 77 54 64 74 76 55 56 63 36 53 62 52 75 46 45
CMT8Y	Route 1:0 49 36 46 8 45 47 11 32 35 34 24 55 39 56 41 43 44 61 16 14 42 15 22 23 25 29 9 33 51 30 20 63 19 48 6 17 106 75 83 96 73 95 82 105 104 103 117 80 102 101 112 79 78 38 64 Route 2:0 97 85 84 66 113 100 99 68 110 111 92 107 74 71 69 115 70 116 81 72 94 93 65 50 76 67 87 88 86 64 Route 3:0 53 109 13 77 59 60 5 37 2 57 114 58 40 98 21 4 54 3 12 26 89 28 91 1 108 52 18 7 31 10 62 90 64 Route 4:0 27 64	Route 1: 0 28 91 64 Route 2: 0 27 52 31 94 108 90 64 Route 3: 0 18 45 8 46 7 48 47 36 49 11 19 62 10 63 32 30 20 51 9 35 34 96 73 107 83 93 95 74 118 75 82 106 104 105 71 72 103 17 69 116 81 70 64 Route 4: 0 53 58 21 40 26 12 76 89 84 98 114 109 64 Route 5: 0 13 6 59 37 5 60 61 16 44 14 42 43 15 57 2 41 22 23 56 39 4 25 55 54 24 29 3 33 1 65 50 67 92 87 110 111 88 68 97 86 112 85 99 66 113 79 101 100 78 38 102 80 117 115 77 64
CMT9X	Route 1:0 38 16 44 43 15 23 25 29 35 20 49 47 8 45 48 7 19 11 32 51 9 34 24 4 39 22 41 42 14 13 61 69 71 94 36 95 96 73 86 84 98 64 58 83 79 78 87 74 66 97 46 63 17 93 70 91 92 55 Route 2:0 88 59 54 75 21 26 80 81 27 28 82 12 67 40 89 72 65 10 30 33 3 50 1 56 31 85 99 52 18 62 60 5 37 6 68 2 57 90 76 77 55 Route 3:0 53 55	Route 1: 0 53 68 100 55 Route 2: 0 27 81 55 Route 3: 0 13 6 18 7 52 31 30 20 32 10 11 19 49 47 48 8 45 5 16 37 44 38 14 42 15 43 92 70 91 69 93 71 60 17 94 63 46 95 36 97 66 73 96 62 99 72 61 55 Route 4: 0 28 12 26 40 2 21 22 41 23 39 25 4 54 24 29 34 35 9 51 33 3 50 1 56 85 65 86 84 74 98 64 87 58 83 78 79 59 88 77 76 90 57 75 89 80 67 82 55
CMT9Y	Route 1:0 49 36 46 8 45 47 11 20 30 51 33 9 29 25 23 41 43 44 61 16 14 42 57 15 22 56 39 55 24 34 35 32 19 7 48 101 38 76 77 99 100 17 70 103 73 81 91 105 94 71 93 80 102 104 62 Route 2:0 95 66 74 108 109 90 50 63 92 106 79 69 72 65 85 86 110 97 98 113 67 78 111 64 96 82 83 84 62 Route 3:0 6 68 52 18 60 5 59 37 75 13 2 21 40 58 112 53 107 4 54 3 12 26 87 28 89 10 31 1 88 27 62	Route 1: 0 27 28 26 12 74 87 89 62 Route 2: 0 13 6 59 37 61 16 44 14 42 43 99 98 76 38 100 78 113 68 62 Route 3: 0 5 60 18 8 45 46 36 49 11 19 47 48 7 52 31 10 32 30 20 51 9 35 34 94 71 105 81 91 93 72 92 69 103 102 80 73 104 70 101 17 67 114 79 106 88 62 Route 4: 0 53 58 40 21 2 57 15 41 22 4 56 23 39 25 55 54 24 29 3 33 1 63 50 65 90 85 108 109 86 95 84 110 66 82 83 97 77 111 64 75 112 96 107 62
CMT10X	Route 1:0 38 14 42 41 22 23 56 39 55 29 65 66 71 9 35 34 24 25 67 15 43 44 16 61 59 17 46 113 91 128 130 131 116 82 101 97 120 96 105 92 102 104 84 115 129 36 112 126 89 87 109 108 88 110 111 73 Route 2:0 95 94 121 119 77 106 132 78 125 81 80 83 103 74 135 127 114 73 Route 3:0 76 33 3 85 54 68 133 93 21 72 4 98 107 123 75 2 57 122 124 5 37 86 79 90 117 49 47 8 45 48 7 19 62 10 11 32 30 70 51 20 63 64 134 99 100 118 73 Route 4:0 27 28 53 58 40 26 12 50 1 69 31 52 18 60 6 13 73	Route 1: 0 26 12 68 3 29 24 54 55 25 39 67 23 56 4 21 72 22 41 2 57 15 43 42 109 110 88 122 75 93 94 108 95 121 77 106 132 97 120 119 96 101 105 76 133 85 98 100 73 Route 2: 0 28 27 52 69 1 50 51 33 34 35 9 71 66 65 130 136 131 82 116 92 102 135 103 74 134 99 73 Route 3: 0 6 59 37 61 16 44 38 14 87 111 89 126 17 112 46 81 114 80 117 90 125 78 124 79 73 Route 4: 0 18 60 5 45 8 47 48 7 10 31 70 30 20 32 63 11 19 62 49 64 129 115 36 113 91 84 128 127 83 104 73 Route 5: 0 53 40 58 13 86 123 107 118 73
CMT10Y	Route 1:0 49 47 45 8 46 36 64 63 65 35 34 24 55 25 39 56 15 43 41 23 67 29 9 33 51 71 66 20 30 32 11 19 48 114 38 17 83 116 115 93 86 106 104 94 133 118 84 103 99 98 107 138 132 130 131 117 75 Route 2:0 134 108 122 78 136 121 79 95 139 123 110 112 113 128 127 80 91 89 111 77 124 90 96 97 75 Route 3:0 129 92 18 60 81 6 88 22 73 21 72 4 74 57 42 14 44 16 61 5 37 59 126 119 82 7 62 85 10 105 70 137 76 50 87 109 120 125 100 54 68 3 135 102 101 75 Route 4:0 53 28 27 52 31 69 1 12 26 40 58 2 13 75	Route 1: 0 28 26 12 87 100 102 75 Route 2: 0 52 18 60 5 6 59 37 61 16 44 14 42 57 15 43 112 90 124 77 111 89 38 113 91 128 17 114 83 127 80 126 81 92 119 101 75 Route 3: 0 27 1 69 31 7 8 45 46 36 47 48 19 49 64 11 62 10 63 32 70 30 20 66 9 33 51 118 84 107 78 135 103 98 121 122 99 79 108 134 123 97 96 110 139 140 95 109 125 88 120 75 Route 4: 0 53 40 58 13 2 41 22 74 72 21 73 4 56 23 67 39 25 55 54 24 29 68 3 34 35 71 65 132 138 133 94 106 104 85 137 50 76 105 136 130 131 117 115 116 93 86 129 82 75

CMT11X	Route 1:0 66 64 61 57 53 52 54 59 50 46 29 49 51 65 62 63 60 56 55 58 120 117 115 114 116 112 108 106 104 101 100 99 34 35 32 28 98 24 95 22 27 30 33 36 97 103 109 111 110 105 102 107 113 121 118 119 122 72 Route 2:0 91 89 88 96 93 92 94 84 80 85 87 82 81 83 86 73 72 Route 3:0 123 67 70 69 125 71 68 124 76 31 19 17 16 26 47 45 40 43 48 42 39 38 37 41 44 21 20 23 25 79 13 8 12 14 11 3 4 10 9 15 78 7 6 5 77 74 2 1 75 90 18 72	Route 1: 0 56 58 55 53 52 54 57 59 61 65 62 64 60 63 66 122 119 120 118 117 121 114 115 116 102 105 107 110 113 112 109 111 108 106 103 104 101 99 100 124 126 125 123 72 Route 2: 0 69 67 70 71 68 40 43 45 48 51 50 46 49 47 44 41 42 39 38 37 29 25 31 19 16 17 26 23 20 21 93 96 94 92 89 88 91 95 22 24 27 28 35 32 97 34 36 33 30 98 72 Route 3: 0 18 3 2 1 4 5 6 7 10 9 11 15 14 13 8 12 84 80 76 75 77 78 79 82 83 87 86 85 81 74 73 90 72
CMT11Y	Route 1:0 65 62 63 64 61 55 53 54 57 51 45 40 43 48 49 47 42 39 38 37 34 25 19 31 36 29 41 44 46 50 59 108 52 109 107 101 96 99 104 105 103 98 95 94 93 35 86 92 27 33 91 97 100 102 106 110 112 58 56 60 113 111 66 Route 2:0 89 22 24 30 28 32 90 26 87 21 20 23 88 82 16 17 83 85 78 12 8 74 79 13 15 81 70 72 6 5 71 68 2 1 69 3 4 76 73 7 75 9 10 11 77 14 80 67 84 18 66	Route 1: 0 46 49 47 44 37 38 39 41 42 48 50 51 45 40 43 65 61 62 64 63 55 53 54 57 59 110 109 52 108 60 113 112 111 114 56 58 96 99 101 107 106 104 103 102 105 100 97 98 95 94 93 83 66 Route 2: 0 16 17 19 25 24 22 20 21 23 26 28 32 36 34 31 30 29 91 35 92 33 27 89 85 82 88 90 86 87 8 12 14 13 15 11 10 9 7 6 5 3 4 2 1 68 67 71 72 73 75 76 70 69 79 78 74 80 81 77 66 Route 3: 0 18 84 66
CMT12X	Route 1:0 70 71 76 82 81 78 77 80 79 73 63 66 62 61 68 64 72 56 60 58 54 57 59 55 53 35 116 117 115 114 130 128 131 139 140 118 120 119 39 38 83 Route 2:0 152 151 156 157 153 150 97 95 102 101 100 96 98 99 83 Route 3:0 159 158 160 133 138 136 135 141 146 142 154 148 125 122 123 126 121 144 137 134 83 Route 4:0 113 30 29 112 36 34 33 32 31 37 132 52 50 45 46 44 48 51 127 149 145 147 143 124 129 107 110 111 109 94 14 15 13 17 18 19 16 12 93 11 91 9 89 90 88 155 86 87 84 1 2 85 92 106 104 103 105 108 83 Route 5:0 28 26 10 4 8 23 22 25 24 27 41 43 47 49 42 40 69 67 65 74 83 Route 6:0 20 21 6 7 5 3 75 83	Route 1: 0 52 44 40 41 42 43 47 49 46 45 48 51 50 59 57 55 54 53 56 58 60 140 138 136 133 134 139 137 135 83 Route 2: 0 67 65 63 69 66 62 74 72 61 64 68 148 144 152 141 146 149 142 154 143 145 147 83 Route 3: 0 10 11 8 9 6 4 2 1 75 3 5 7 90 86 88 155 84 85 87 89 91 92 94 93 109 111 113 112 110 108 107 103 104 106 105 83 Route 4: 0 82 81 79 80 77 78 76 71 73 70 153 150 151 156 158 161 159 157 160 83 Route 5: 0 20 21 22 23 26 28 30 24 25 27 29 31 35 37 33 32 34 36 119 117 39 120 38 118 114 116 115 130 127 129 132 131 128 126 124 123 125 121 122 83 Route 6: 0 19 18 17 13 15 16 14 12 95 97 99 98 96 100 101 102 83
CMT12Y	Route 1:0 14 12 19 18 37 36 34 33 32 53 55 48 46 44 50 31 35 38 17 13 15 16 70 72 73 71 75 39 94 91 90 102 89 93 95 76 77 74 58 Route 2:0 108 110 109 106 104 107 87 85 82 83 81 69 84 86 88 92 105 103 100 98 99 97 96 101 58 Route 3:0 40 54 57 56 79 78 30 27 25 26 28 29 80 59 1 64 6 9 8 66 68 10 11 67 62 4 65 7 63 5 3 61 2 60 58 Route 4:0 51 45 41 42 43 47 49 52 24 22 23 20 21 58	Route 1: 0 2 1 4 3 5 7 6 9 8 10 11 17 13 18 19 15 16 14 12 70 72 74 73 71 75 77 76 66 68 69 67 64 65 63 61 62 60 59 58 Route 2: 0 21 20 22 23 26 28 30 27 29 25 24 32 33 31 35 38 37 36 34 92 94 39 95 93 89 91 90 87 88 86 85 82 83 84 81 80 79 78 58 Route 3: 0 43 42 41 40 44 45 46 48 51 50 52 47 49 57 55 54 56 53 108 111 109 110 96 97 98 100 101 103 106 105 107 104 102 99 58
CMT13X	Route 1:0 63 60 62 61 64 56 55 59 50 46 31 44 42 47 49 48 45 51 57 54 52 53 58 107 108 106 100 95 98 103 104 102 97 94 93 92 34 35 32 28 91 24 88 85 22 27 30 33 36 90 96 99 101 105 109 110 111 65 Route 2:0 84 82 81 89 86 87 77 73 78 80 69 75 76 79 66 65 Route 3:0 18 83 74 19 17 16 26 21 20 23 41 43 40 39 38 37 29 25 72 13 8 12 14 11 3 4 10 9 15 71 7 6 5 70 67 2 1 68 65	Route 1: 0 52 59 57 54 53 55 58 56 63 60 64 62 61 110 111 112 108 107 109 95 98 100 103 106 105 102 104 101 99 96 92 93 97 94 65 Route 2: 0 18 4 3 5 6 7 9 10 11 15 14 13 8 12 2 1 66 67 68 69 70 71 72 74 75 76 80 79 78 77 73 83 65 Route 3: 0 43 40 45 48 51 50 49 46 44 47 41 42 39 37 38 26 23 21 20 17 16 19 25 31 29 90 32 28 91 35 36 34 33 30 27 22 24 88 84 87 85 89 86 82 81 65
CMT13Y	Route 1:0 66 64 61 55 53 54 57 59 50 46 31 44 42 39 38 41 47 49 48 43 40 45 51 65 62 63 75 130 60 56 58 129 132 124 118 113 116 121 122 120 115 112 111 110 107 103 35 114 117 119 123 127 125 52 126 128 131 133 83 Route 2:0 33 27 102 100 99 104 105 106 108 109 95 91 96 98 87 89 88 85 86 93 92 94 97 84 83 Route 3:0 101 34 30 24 22 16 17 19 25 23 21 20 26 28 32 37 29 36 12 8 13 9 90 147 80 79 146 77 144 68 135 140 73 76 143 145 78 139 137 70 134 67 69 136 138 71 74 141 72 142 83 Route 4:0 14 15 11 3 2 1 5 6 4 7 10 18 81 82 83	Route 1: 0 37 38 39 41 42 40 43 45 47 44 46 49 50 51 48 55 53 54 57 59 61 65 62 64 63 66 133 131 130 60 56 58 129 132 128 127 126 125 52 112 111 110 114 115 117 119 122 120 123 124 121 118 116 113 83 Route 2: 0 26 23 21 20 17 16 19 22 24 25 28 31 30 34 36 32 29 108 35 109 33 27 106 102 99 100 105 107 104 103 83 Route 3: 0 75 72 71 74 70 67 69 73 76 68 77 78 79 80 147 146 135 140 143 144 145 142 139 141 138 134 137 136 83 Route 4: 0 82 81 18 101 148 83 Route 5: 0 8 12 13 14 15 11 10 9 7 6 4 3 5 2 1 85 84 86 87 88 89 90 92 93 94 98 97 96 95 91 83
CMT14X	Route 1:0 37 36 19 18 17 13 15 14 12 16 34 33 32 51 48 45 50 31 35 53 55 59 57 54 58 60 56 63 123 127 125 121 124 126 122 120 105 38 84 70 Route 2:0 107 39 106 89 82 86 104 102 118 119 115 113 108 109 110 112 117 101 103 70 Route 3:0 133 114 111 116 99 94 97 96 93 98 100 88 87 81 80 78 76 77 75 73 74 71 72 79 83 85 70 Route 4:0 128 130 134 132 62 66 129 64 40 46 43 47 49 52 42 41 44 68 61 131 92 90 91 30 29 27 28 26 25 24 95 70 Route 5:0 1 6 8 23 21 69 67 65 20 22 10 11 9 4 7 5 3 2 70	Route 1: 0 63 65 67 62 66 69 61 64 68 59 57 55 54 53 56 58 60 127 125 123 120 121 122 124 126 131 135 128 129 133 134 130 132 70 Route 2: 0 20 21 22 23 26 28 30 29 27 25 24 94 97 95 93 92 91 90 99 100 98 96 71 72 74 73 75 77 76 78 79 81 80 70 Route 3: 0 41 40 43 42 44 46 45 48 51 50 52 49 47 36 34 32 33 31 37 35 105 38 107 39 106 104 103 102 101 119 116 114 111 113 115 118 117 112 110 108 109 70 Route 4: 0 5 3 7 1 2 4 6 8 9 11 10 13 17 18 19 15 16 14 12 82 84 86 85 83 88 89 87 70
CMT14Y	Route 1:0 14 12 19 18 37 36 34 33 32 53 55 59 44 48 31 35 38 17 13 15 16 72 74 75 73 77 39 96 93 92 104 91 95 97 78 79 76 60 Route 2:0 113 112 114 111 110 103 98 99 101 100 102 105 106	Route 1: 0 20 21 22 23 26 28 30 29 27 25 24 17 13 18 19 15 16 14 12 72 74 76 75 73 77 79 78 84 85 87 89 90 88 86 83 82 81 80 60 Route 2: 0 40 41 42 43 47 49 52 50 51 48 45 44 46 55 57 59 54 53 56 58 115 113 110 111 114 112 99 98 100 102 103 105

	107 94 90 87 84 86 61 88 89 109 108 60 Route 3:0 71 11 10 70 68 8 67 7 65 3 63 4 64 6 9 66 2 1 62 69 83 81 80 82 26 28 30 29 27 85 56 57 54 58 60 Route 4:0 51 52 49 47 43 46 42 41 40 45 50 25 23 5 21 20 22 24 60	108 107 109 106 104 101 60 Route 3: 0 10 11 9 8 6 7 5 3 4 1 2 62 61 67 65 63 64 66 69 68 71 70 34 36 32 33 37 38 35 31 91 95 97 39 96 94 93 92 60

Appendix B: Run three of VRPDDP

Instance	Initial Solution Routes	Final Solution Routes
CMT1X	Route 1:0 36 3 74 77 72 43 26 31 28 53 20 75 67 80 76 10 84 45 79 33 39 60 30 34 94 85 90 61 59 65 95 21 29 35 81 2 68 58 57 69 70 88 51 Route 2:0 46 9 40 41 4 62 71 7 8 52 1 93 23 24 14 63 25 13 86 42 89 82 54 17 37 55 83 49 64 15 44 87 19 12 92 66 56 73 78 91 51 Route 3:0 27 6 48 22 16 5 11 32 38 50 18 47 51	Route 1: 0 73 68 3 53 20 36 81 35 2 78 51 Route 2: 0 12 38 59 39 84 79 94 49 55 5 83 50 65 80 30 9 95 21 75 67 29 16 34 76 10 60 11 46 51 Route 3: 0 91 51 Route 4: 0 47 54 18 66 25 13 41 86 19 62 92 14 71 4 85 40 17 37 82 64 89 33 45 90 15 87 42 44 61 51 Route 5: 0 63 6 56 93 48 27 69 70 24 43 88 23 57 7 72 8 58 26 77 28 74 31 22 52 1 32 51
CMT1Y	Route 1:0 40 41 4 62 43 7 26 23 24 25 13 86 42 44 15 90 79 84 10 76 9 16 29 80 50 75 67 36 21 81 34 95 59 30 60 39 33 45 64 89 87 19 85 18 70 69 88 51 Route 2:0 46 49 37 17 12 5 38 11 65 20 53 22 32 1 68 74 77 8 58 48 6 57 72 31 28 3 2 35 52 47 54 63 71 51 Route 3:0 83 55 94 82 61 92 66 14 56 93 73 78 27 91 51	Route 1: 0 25 71 24 70 57 7 88 43 69 23 6 14 63 56 93 48 26 77 72 8 58 51 Route 2: 0 73 27 51 Route 3: 0 61 84 39 10 80 38 11 83 9 16 65 5 55 64 89 37 82 44 45 15 33 90 79 60 49 94 51 Route 4: 0 78 32 2 52 1 74 31 22 68 28 53 81 36 3 35 20 29 75 76 30 34 95 67 21 59 50 51 Route 5: 0 92 47 18 66 13 62 41 86 40 19 85 42 87 17 4 54 12 91 46 51
CMT2X	Route 1:0 60 66 139 105 129 9 72 99 50 124 24 49 23 116 56 92 18 25 55 31 65 112 140 133 59 88 128 144 93 109 127 85 11 132 113 84 71 117 76 Route 2:0 53 10 39 32 130 115 145 61 64 41 98 106 83 58 38 82 14 134 111 110 143 94 76 Route 3:0 54 57 70 69 138 43 97 42 96 22 62 28 135 36 121 95 21 47 80 5 89 15 131 37 20 87 101 67 123 78 125 81 90 107 137 76 Route 4:0 1 136 74 122 103 19 35 13 126 27 119 29 48 104 146 77 102 73 63 3 118 114 44 86 16 33 142 147 79 108 120 141 100 91 76 Route 5:0 7 8 46 52 45 30 2 68 4 76 Route 6:0 34 26 12 40 17 51 6 75 76	Route 1: 0 48 5 47 80 69 110 36 60 70 20 111 71 103 29 119 27 101 46 34 108 8 76 Route 2: 0 53 66 133 85 139 11 140 127 109 88 35 7 120 82 141 76 Route 3: 0 77 2 30 104 52 128 19 87 54 59 93 126 13 57 15 131 89 121 122 45 76 Route 4: 0 14 12 86 40 113 99 129 124 50 25 55 92 18 118 138 64 76 Route 5: 0 97 23 130 41 135 61 22 116 117 43 137 63 90 75 68 142 79 147 4 67 76 Route 6: 0 94 37 144 134 143 95 21 102 74 146 28 96 136 145 62 107 33 73 1 42 115 56 76 Route 7: 0 17 91 114 106 32 83 9 39 132 58 72 84 31 105 10 112 65 38 76 Route 8: 0 6 125 51 81 16 123 49 98 24 78 44 3 100 26 76
CMT2Y	Route 1:0 66 14 54 38 10 139 105 129 99 50 124 98 49 23 117 42 130 92 18 25 55 31 65 140 133 59 85 112 11 127 88 134 76 Route 2:0 15 60 69 135 22 138 41 97 6 145 43 115 116 64 96 61 143 110 36 71 144 94 89 111 37 20 80 131 70 128 67 72 109 93 87 121 146 76 Route 3:0 56 123 24 63 78 90 137 1 62 102 95 122 103 57 101 47 21 28 136 77 68 73 107 33 118 113 106 76 Route 4:0 19 13 48 5 8 82 35 84 132 9 114 44 16 125 91 3 32 83 39 86 141 26 58 53 126 119 104 147 76 Route 5:0 74 29 27 46 7 120 108 34 52 45 30 2 75 79 4 142 81 51 40 12 100 76 Route 6:0 17 76	Route 1: 0 26 86 51 91 125 40 12 58 10 65 66 139 38 72 132 84 105 31 76 Route 2: 0 30 104 119 45 5 121 110 37 20 57 131 15 89 80 47 36 21 134 111 94 144 60 70 71 69 143 76 Route 3: 0 120 112 140 11 85 59 133 88 14 109 53 127 82 8 46 76 Route 4: 0 100 17 114 9 113 83 39 25 99 92 18 129 55 118 78 32 3 98 24 50 124 44 106 76 Route 5: 0 16 90 49 123 130 97 56 117 115 41 138 64 42 116 23 22 96 61 1 76 Route 6: 0 77 33 43 62 73 137 63 145 107 2 75 68 142 147 76 Route 7: 0 67 76 Route 8: 0 79 126 128 87 54 19 93 35 7 141 34 52 27 13 101 122 48 29 103 108 4 76 Route 9: 0 95 74 102 136 146 135 28 6 81 76
CMT3X	Route 1:0 38 55 37 57 5 6 18 8 17 16 14 56 2 21 22 63 74 4 65 64 69 73 9 60 72 10 52 7 19 36 11 51 32 70 42 20 50 35 34 44 29 24 25 45 39 23 62 15 43 13 46 49 48 59 41 Route 2:0 61 75 40 26 66 68 67 27 28 12 53 3 33 30 1 71 31 58 47 54 41	Route 1: 0 28 27 67 42 53 29 44 69 3 1 31 7 11 59 19 36 72 70 71 48 51 10 52 32 60 30 20 35 34 73 50 9 33 12 68 41 Route 2: 0 21 61 2 43 15 56 63 23 22 62 74 65 25 39 4 45 64 24 66 26 41 Route 3: 0 13 54 6 47 18 58 5 8 49 46 17 16 38 55 14 57 37 75 40 41
CMT3Y	Route 1:0 6 52 31 49 36 101 47 45 44 43 57 2 42 14 16 8 46 99 7 77 70 10 90 30 78 34 87 82 55 106 56 83 23 94 15 41 107 39 25 24 29 62 91 35 68 9 102 51 88 20 32 11 19 100 98 75 17 67 48 38 73 59 Route 2:0 96 97 5 18 64 37 95 61 108 74 22 80 79 58 53 40 21 4 105 33 69 66 89 1 60 50 3 12 28 27 26 71 54 63 92 81 13 76 65 72 93 104 85 59 Route 3:0 109 84 86 103 59	Route 1: 0 86 72 13 6 109 65 5 17 64 16 44 97 75 37 73 38 14 43 74 15 96 95 42 108 2 57 79 61 93 58 40 21 59 Route 2: 0 81 107 56 94 22 80 41 23 92 39 55 83 25 54 24 106 4 63 105 103 52 59 Route 3: 0 101 49 36 98 45 8 47 99 19 70 11 10 89 69 77 66 100 67 46 48 7 76 18 59 Route 4: 0 60 50 1 31 30 51 9 68 102 20 88 90 32 78 91 35 62 33 34 3 87 29 82 12 71 26 84 53 104 28 85 27 59
CMT4X	Route 1:0 38 65 5 7 20 18 37 42 88 66 86 72 2 21 22 73 4 35 34 84 75 29 74 83 24 25 39 23 41 15 43 89 67 45 36 47 48 69 11 19 49	Route 1: 0 81 68 18 90 56 67 89 5 57 13 51 Route 2: 0 71 72 4 39 84 55 26 76 63 12 29 79 24 74 75 25 73 23 41 86 22 21 40 2 53 42 87 88 15 43 66 14 44 65 38 16 37 64 6

	46 8 90 17 16 44 14 87 13 93 91 58 51 Route 2:0 32 9 79 55 53 71 85 40 26 76 78 77 27 28 12 63 3 54 50 52 1 33 60 70 80 30 82 62 59 68 57 64 6 56 92 10 61 81 31 51	85 28 27 51 Route 3: 0 92 19 69 7 31 82 32 93 62 11 91 46 49 36 47 48 8 59 45 17 58 10 61 30 20 70 50 3 54 33 9 60 83 34 35 80 1 52 77 78 51
CMT4Y	Route 1:0 6 23 41 39 55 24 25 67 7 49 127 64 113 36 47 45 44 43 15 14 16 8 46 111 89 62 10 82 126 102 30 20 114 9 90 129 66 128 34 80 35 103 65 32 63 11 19 132 78 112 110 87 17 79 48 38 107 108 119 118 71 Route 2:0 31 69 68 12 1 70 51 60 18 5 109 85 2 21 86 130 95 104 4 29 74 33 3 99 94 54 117 75 56 93 22 92 106 57 73 120 42 37 124 61 76 122 59 77 123 125 133 72 81 71 Route 3:0 88 115 52 101 100 50 83 131 91 105 96 98 97 27 28 26 40 121 116 53 58 84 13 71	Route 1: 0 117 99 29 74 131 3 50 102 66 129 65 128 35 34 103 9 33 63 126 32 20 90 72 1 132 69 27 28 71 Route 2: 0 41 106 92 93 130 23 104 25 95 67 39 119 91 56 22 57 120 2 73 107 108 43 42 15 86 38 14 71 Route 3: 0 89 19 47 36 11 82 64 127 49 113 30 114 51 80 100 10 81 31 133 70 101 7 78 112 46 8 45 79 48 111 125 62 18 88 115 52 97 71 Route 4: 0 26 96 110 60 5 122 59 123 6 77 76 87 17 61 124 16 85 109 44 37 53 71 Route 5: 0 98 58 84 13 40 105 116 121 12 83 68 4 55 54 24 94 118 75 21 71
CMT5X	Route 1:0 38 112 88 42 2 57 111 89 67 25 24 55 39 56 23 41 15 43 14 44 90 127 61 45 36 114 48 92 85 66 129 105 35 106 131 137 65 32 63 11 19 47 116 130 64 49 46 113 17 16 110 123 59 133 121 98 74 Route 2:0 54 34 132 93 103 136 70 31 84 128 8 18 60 37 5 82 115 7 62 10 30 20 51 33 117 9 83 71 29 107 122 95 96 97 102 134 77 3 120 68 78 94 139 109 74 Route 3:0 22 73 21 138 4 72 76 125 80 6 126 81 104 69 100 27 135 75 12 28 101 99 108 124 119 53 58 40 26 86 50 1 52 118 91 79 13 87 74	Route 1: 0 91 52 118 31 136 10 63 129 64 11 62 114 19 47 115 92 7 128 6 87 13 108 58 40 26 119 124 74 Route 2: 0 99 78 134 68 86 12 54 120 97 25 121 72 21 73 138 4 122 23 56 39 107 55 98 67 133 96 95 139 94 76 123 42 110 2 41 57 15 89 109 22 74 Route 3: 0 80 127 126 60 5 18 82 8 61 90 16 112 79 45 113 17 125 59 37 44 14 88 111 43 38 74 Route 4: 0 100 81 48 46 36 116 49 130 85 104 70 103 30 84 51 83 106 34 9 33 117 3 24 102 29 77 1 50 75 69 135 27 74 Route 5: 0 53 101 28 71 131 65 35 137 132 66 93 20 32 105 74
CMT5Y	Route 1:0 49 46 19 11 63 64 36 38 44 100 37 42 123 101 57 121 73 54 80 4 108 23 134 22 74 75 56 119 110 133 29 65 34 24 55 25 39 145 67 41 15 43 14 124 102 45 17 16 139 61 161 85 128 142 95 86 Route 2:0 6 47 82 8 126 62 97 10 32 30 20 105 144 66 149 118 81 129 51 33 89 156 109 114 79 78 35 9 71 143 117 96 140 141 104 127 125 160 138 94 158 93 7 48 159 91 122 152 135 86 Route 3:0 84 83 103 52 18 60 5 137 59 107 151 88 2 40 136 13 58 120 106 21 72 132 68 77 154 115 148 116 31 70 1 147 112 69 87 50 76 155 3 146 12 28 111 26 98 157 90 150 153 86 Route 4:0 99 92 130 27 113 53 131 86	Route 1: 0 116 31 147 87 69 1 70 148 81 115 32 105 20 143 144 66 117 30 19 104 93 7 11 63 62 140 141 97 142 64 10 96 86 Route 2: 0 67 145 39 23 108 56 150 72 153 75 134 119 110 25 55 133 132 54 4 90 107 121 22 152 74 41 151 73 21 88 57 106 120 58 40 86 Route 3: 0 53 136 98 26 28 113 111 80 155 89 79 114 156 154 76 3 109 29 24 77 157 12 68 146 34 149 65 71 51 33 118 35 78 9 129 95 50 27 112 86 Route 4: 0 103 18 158 125 46 126 48 49 128 36 47 8 82 127 130 52 86 Route 5: 0 2 135 101 15 43 123 100 38 44 14 124 60 91 159 138 5 160 84 94 45 83 17 16 61 85 59 161 102 139 42 122 37 137 13 99 92 6 131 86
CMT6X	Route 1:0 36 3 74 77 72 43 26 31 28 53 20 75 67 80 76 10 84 45 79 33 39 60 30 34 94 85 90 61 59 65 95 21 29 35 81 2 68 58 57 69 70 88 51 Route 2:0 46 9 40 41 4 62 71 7 8 52 1 93 23 24 14 63 25 13 86 42 89 82 54 17 37 55 83 49 64 15 44 87 19 12 92 66 56 73 78 91 51 Route 3:0 27 6 48 22 16 5 11 32 38 50 18 47 51	Route 1: 0 56 88 43 24 63 70 57 23 69 7 95 29 67 21 75 65 50 16 38 83 5 11 2 78 32 27 73 51 Route 2: 0 66 18 92 14 71 25 54 62 13 41 40 85 42 87 19 86 4 47 51 Route 3: 0 90 10 60 84 39 94 49 59 34 80 9 76 30 33 79 45 64 37 15 44 89 82 17 61 55 12 51 Route 4: 0 48 93 58 26 72 8 77 52 68 3 20 36 35 81 53 31 74 28 22 1 51 Route 5: 0 6 91 46 51
CMT6Y	Route 1:0 40 41 4 62 43 7 26 23 24 25 13 86 42 44 15 90 79 84 10 76 9 16 29 80 50 75 67 36 21 81 34 95 59 30 60 39 33 45 64 89 87 19 85 18 70 69 88 51 Route 2:0 46 49 37 17 12 5 38 11 65 20 53 22 32 1 68 74 77 8 58 48 6 57 72 31 28 3 2 35 52 47 54 63 71 51 Route 3:0 83 55 94 82 61 92 66 14 56 93 73 78 27 91 51	Route 1: 0 63 24 57 23 69 70 6 56 93 48 58 52 1 68 22 74 77 31 26 72 8 7 43 88 42 87 19 51 Route 2: 0 92 82 17 54 47 37 15 64 89 44 4 18 14 25 71 66 62 40 85 13 41 86 51 Route 3: 0 73 32 78 27 2 3 53 20 35 81 36 28 11 46 51 Route 4: 0 91 61 12 51 Route 5: 0 65 16 83 59 9 94 49 45 90 79 33 39 84 10 60 76 30 50 95 21 80 34 67 29 75 38 55 5 51
CMT7X	Route 1:0 60 66 139 105 129 9 72 99 50 124 24 49 23 116 56 92 18 25 55 31 65 112 140 133 59 88 128 144 93 109 127 85 11 132 113 84 71 117 76 Route 2:0 53 10 39 32 130 115 145 61 64 41 98 106 83 58 38 82 14 134 111 110 143 94 76 Route 3:0 54 57 70 69 138 43 97 42 96 22 62 28 135 36 121 95 21 47 80 5 89 15 131 37 20 87 101 67 123 78 125 81 90 107 137 76 Route 4:0 1 136 74 122 103 19 35 13 126 27 119 29 48 104 146 77 102 73 63 3 118 114 44 86 16 33 142 147 79 108 120 141 100 91 76 Route 5:0 7 8 46 52 45 30 2 68 4 76 Route 6:0 34 26 12 40 17 51 6 75 76	Route 1: 0 95 48 103 122 144 134 70 60 71 69 36 47 110 143 62 145 107 1 73 76 Route 2: 0 83 9 105 31 72 113 39 84 58 132 38 65 66 140 139 11 112 76 Route 3: 0 90 16 33 137 63 49 123 23 130 56 97 24 98 40 3 125 76 Route 4: 0 147 29 80 5 121 37 94 111 20 15 57 89 27 87 52 14 82 54 128 13 131 101 76 Route 5: 0 100 141 79 2 119 30 104 45 46 108 34 76 Route 6: 0 10 114 12 86 32 44 78 118 124 106 50 99 18 92 129 55 25 91 17 76 Route 7: 0 142 68 75 51 81 6 77 74 21 102 136 146 28 22 61 135 96 116 138 64 41 43 115 42 117 76 Route 8: 0 7 127 109 93 133 19 88 59 85 53 35 8 120 126 4 76 Route 9: 0 67 26 76

CMT7Y	Route 1:0 66 14 54 38 10 139 105 129 99 50 124 98 49 23 117 42 130 92 18 25 55 31 65 140 133 59 85 112 11 127 88 134 76 Route 2:0 15 60 69 135 22 138 41 97 6 145 43 115 116 64 96 61 143 110 36 71 144 94 89 111 37 20 80 131 70 128 67 72 109 93 87 121 146 76 Route 3:0 56 123 24 63 78 90 137 1 62 102 95 122 103 57 101 47 21 28 136 77 68 73 107 33 118 113 106 76 Route 4:0 19 13 48 5 8 82 35 84 132 9 114 44 16 125 91 3 32 83 39 86 141 26 58 53 126 119 104 147 76 Route 5:0 74 29 27 46 7 120 108 34 52 45 30 2 75 79 4 142 81 51 40 12 100 76 Route 6:0 17 76	Route 1: 0 100 114 40 132 58 86 12 39 113 72 31 105 99 55 129 18 124 3 78 44 118 17 91 76 Route 2: 0 26 7 19 93 54 82 35 8 109 127 88 59 133 14 128 131 57 87 13 101 27 103 76 Route 3: 0 140 66 85 11 65 83 9 84 38 53 139 10 112 76 Route 4: 0 79 104 122 30 4 147 76 Route 5: 0 141 108 126 120 34 52 46 67 76 Route 6: 0 6 102 28 77 2 62 136 21 95 61 135 33 96 138 64 117 130 56 137 107 145 1 73 22 146 74 76 Route 7: 0 32 106 25 50 92 98 49 123 24 23 97 116 42 41 43 115 76 Route 8: 0 45 119 29 48 121 80 5 89 15 37 111 20 94 144 143 47 71 134 70 60 69 36 110 76 Route 9: 0 81 68 142 75 125 51 90 16 63 76
CMT8X	Route 1:0 38 59 5 11 20 9 32 19 37 42 82 60 80 66 2 21 22 67 78 4 69 68 73 77 54 35 64 34 48 29 24 25 49 39 23 41 15 81 43 83 61 36 8 17 16 50 44 14 51 13 56 55 76 63 52 46 45 Route 2:0 10 74 3 57 47 65 79 6 58 40 26 70 72 71 27 28 12 33 30 1 75 53 62 18 7 31 45	Route 1: 0 1 46 30 32 76 64 20 74 75 31 55 5 17 61 16 44 83 38 14 59 37 81 42 13 6 51 71 72 27 45 Route 2: 0 8 53 7 52 18 62 50 56 11 63 36 19 10 9 35 33 54 77 34 3 48 29 24 68 73 45 Route 3: 0 47 67 22 23 41 66 80 49 25 4 39 78 69 57 12 28 26 70 40 79 21 65 2 15 60 43 82 58 45
CMT8Y	Route 1:0 37 61 60 7 49 36 106 47 45 44 43 15 23 41 42 14 16 8 46 104 82 62 10 75 119 95 30 83 34 92 87 39 88 55 25 24 29 96 35 73 9 107 51 93 20 32 63 11 19 71 105 103 80 17 72 48 38 100 101 112 111 64 Route 2:0 28 53 102 5 18 52 1 31 116 69 117 78 57 66 58 13 2 113 79 86 56 84 26 40 21 68 110 12 67 33 65 50 3 54 4 97 22 85 99 59 6 115 81 118 94 74 90 64 Route 3:0 70 77 114 109 98 89 76 91 27 108 64	Route 1: 0 13 2 40 26 89 50 93 1 65 91 28 12 92 34 96 3 51 20 83 30 107 33 73 9 35 76 90 64 Route 2: 0 6 115 37 14 78 102 16 17 61 59 69 116 60 5 80 38 44 117 64 Route 3: 0 114 58 66 77 101 43 79 15 100 42 57 113 85 99 22 41 86 23 112 56 25 88 111 55 110 54 68 39 4 21 98 84 97 24 87 29 67 64 Route 4: 0 27 108 94 106 104 47 82 19 48 7 71 105 103 8 45 72 46 36 49 11 75 63 62 118 74 32 119 95 10 31 52 81 18 70 109 53 64
CMT9X	Route 1:0 38 69 5 7 6 37 42 92 70 90 76 2 35 21 22 77 4 34 54 88 79 29 78 87 24 25 39 23 41 15 43 93 71 45 36 47 48 73 11 19 49 46 8 94 17 16 44 14 91 13 97 95 62 55 Route 2:0 32 30 84 74 51 83 59 57 68 53 40 89 75 101 67 28 81 82 100 80 26 12 3 58 50 56 1 33 98 64 9 20 86 66 63 72 99 52 18 61 60 96 10 65 85 31 55 Route 3:0 27 55	Route 1: 0 81 60 5 18 52 99 72 7 65 31 85 62 94 96 8 63 46 45 71 17 16 38 93 69 70 91 42 15 92 43 44 14 37 2 89 68 57 13 6 61 100 55 Route 2: 0 23 77 88 39 79 25 4 59 76 41 22 90 21 40 26 101 75 80 53 55 Route 3: 0 36 97 19 11 66 73 49 47 95 48 10 32 86 84 30 74 98 20 51 9 35 87 34 33 64 83 29 78 24 54 67 12 58 3 50 1 56 27 55 Route 4: 0 28 82 55
CMT9Y	Route 1:0 6 31 7 49 36 104 47 45 44 43 57 42 14 61 16 8 46 102 80 73 10 93 30 81 34 90 85 55 109 56 86 23 97 15 41 110 39 25 24 29 65 94 35 71 9 105 51 91 20 32 11 19 103 101 78 17 70 48 38 76 62 Route 2:0 27 53 13 115 67 114 79 72 69 52 18 60 5 113 37 100 99 98 111 64 2 58 40 26 82 21 83 22 4 108 33 1 63 50 3 12 54 66 95 84 77 59 75 68 92 106 88 28 107 62 Route 3:0 112 96 87 74 89 62	Route 1: 0 28 89 75 6 13 37 113 101 45 67 114 68 59 115 78 16 42 98 99 43 76 38 100 14 44 61 17 5 60 18 79 106 52 27 62 Route 2: 0 82 4 97 83 41 22 23 110 56 84 95 55 109 39 108 12 74 29 90 85 24 54 86 25 66 107 53 62 Route 3: 0 33 9 105 71 3 65 94 34 35 26 21 112 57 111 2 15 77 64 58 40 96 87 88 62 Route 4: 0 104 102 70 103 8 46 69 7 48 19 80 49 36 47 11 73 72 10 91 93 32 20 81 51 30 63 50 31 92 1 62
CMT10X	Route 1:0 38 111 87 42 2 57 110 88 67 25 24 55 39 56 23 41 15 43 14 44 89 126 61 45 36 113 48 91 84 66 128 104 35 105 130 136 65 32 63 11 19 47 115 129 64 49 46 112 17 16 109 122 59 132 120 97 73 Route 2:0 54 34 131 92 102 135 70 31 83 127 8 18 60 37 5 81 114 7 62 10 134 30 20 51 33 116 9 82 71 29 106 121 94 95 96 101 133 76 3 119 68 77 93 75 108 79 73 Route 3:0 22 137 21 4 72 13 86 6 78 90 117 52 1 12 28 99 27 100 98 107 123 118 53 58 40 26 85 50 74 69 103 80 125 124 73	Route 1: 0 67 132 106 120 55 77 93 22 72 21 137 4 56 121 39 23 95 94 108 41 15 88 122 57 73 Route 2: 0 71 66 131 130 65 35 105 34 82 9 136 92 51 20 102 30 135 70 10 104 32 83 134 1 74 69 28 99 27 100 26 107 53 118 73 Route 3: 0 50 85 133 54 24 96 25 97 119 116 3 76 33 29 101 68 12 98 75 2 40 58 123 73 Route 4: 0 44 111 16 89 61 5 78 60 17 8 81 7 117 90 125 6 126 38 14 43 109 42 110 87 124 59 37 86 79 13 73 Route 5: 0 84 11 48 47 114 80 113 115 18 45 112 36 46 49 64 129 128 63 62 19 91 127 31 103 52 73
CMT10Y	Route 1:0 25 9 7 49 131 64 117 36 47 45 44 15 41 42 6 16 8 46 115 93 62 10 86 130 30 106 132 66 20 138 34 67 55 24 107 35 84 71 94 133 65 32 63 11 19 116 114 91 111 43 17 83 48 38 112 110 122 123 75 Route 2:0 60 2 73 40 3 51 70 33 68 21 72 57 113 89 134 108 99 39 4 29 103 98 54 79 56 96 141 139 74 22 97 23 90 14 61 80 127 5 128 37 77 124 126 59 129 82 137 76 78 118 85 75 Route 3:0 140 121 12 28 101 27 102 100 26 87 50 1 136 69 105 31 104 135 95 109 125 120 53 58 13 18 119 52 92 81 88 75	Route 1: 0 28 26 53 120 75 Route 2: 0 6 81 113 38 44 128 5 60 80 61 17 114 8 83 45 37 112 43 126 59 91 16 14 89 42 90 111 15 57 124 77 58 13 75 Route 3: 0 102 12 50 66 133 65 35 107 29 103 3 78 98 135 76 70 31 137 85 10 105 69 136 1 33 118 9 34 84 51 71 138 132 104 20 94 30 75 Route 4: 0 40 125 109 100 95 73 21 139 72 56 97 108 55 122 24 68 87 134 67 121 54 99 25 39 79 4 123 22 74 140 141 96 23 110 41 2 88 75 Route 5: 0 11 86 63 129 7 116 82 115 49 117 47 48 46 36 19 93 62 106 32 130 131 64 18 127 92 119 52 27 101 75

CMT11X	Route 1:0 66 122 119 120 118 117 116 115 114 52 53 54 57 59 113 107 110 112 111 108 29 106 104 103 109 46 49 50 101 105 51 34 102 121 65 36 32 100 61 35 99 62 64 63 60 56 55 58 97 30 24 95 22 27 28 98 33 72 Route 2:0 71 69 67 70 68 48 44 42 39 38 37 25 94 21 93 92 20 96 26 23 91 88 84 85 86 87 4 6 18 5 1 2 3 76 10 81 7 9 11 15 14 13 80 8 12 89 17 16 78 19 31 41 47 45 43 40 127 125 123 126 124 90 72 Route 3:0 74 83 79 82 77 75 73 72	Route 1: 0 13 80 12 84 8 87 1 2 74 73 76 77 82 85 83 86 11 9 81 10 7 78 5 4 3 75 15 14 6 79 72 Route 2: 0 123 125 126 70 69 67 71 124 127 68 55 58 53 57 65 121 122 63 56 116 59 115 54 114 52 72 Route 3: 0 42 100 39 99 104 103 41 38 37 108 111 46 47 106 44 49 109 48 51 50 110 112 113 43 101 40 102 107 45 105 61 64 118 120 62 60 119 66 117 72 Route 4: 0 89 17 91 19 16 88 20 94 23 21 93 26 96 92 30 98 31 95 25 24 22 27 33 34 97 29 32 36 35 28 90 18 72
CMT11Y	Route 1:0 29 36 19 25 34 65 61 111 112 114 113 58 55 56 60 63 64 62 115 51 106 105 102 100 37 97 98 40 95 38 94 39 42 41 31 35 44 103 104 101 99 43 45 48 47 46 49 50 107 110 109 108 52 53 54 57 96 91 59 33 93 66 Route 2:0 30 24 85 82 83 17 16 88 21 87 86 20 90 26 23 89 22 27 28 32 92 12 79 80 81 77 9 6 18 72 7 75 76 70 5 1 68 2 69 3 4 73 10 71 11 15 14 13 78 74 8 67 66 Route 3:0 84 66	Route 1: 0 81 67 74 12 8 78 73 75 10 9 7 72 11 76 70 71 5 4 77 13 79 15 80 14 6 68 2 1 69 3 84 18 66 Route 2: 0 16 83 17 89 82 85 87 19 88 90 21 26 20 23 86 22 24 25 33 30 27 29 91 31 34 92 35 32 36 28 66 Route 3: 0 96 56 112 57 110 108 54 53 109 60 113 63 64 61 62 115 65 59 111 114 58 55 52 95 39 100 41 93 99 43 40 101 45 104 48 46 103 49 105 44 47 102 42 98 50 106 107 51 37 38 94 97 66
CMT12X	Route 1:0 34 64 62 68 61 70 151 71 156 158 161 162 82 81 78 76 153 157 159 160 80 79 77 73 150 63 72 133 55 59 37 36 35 57 54 136 138 60 140 58 56 135 53 152 83 Route 2:0 16 12 14 15 13 17 18 19 38 119 33 32 51 48 44 40 46 45 50 31 120 118 134 139 137 141 148 144 66 149 69 146 142 154 74 128 130 131 39 102 83 Route 3:0 49 41 47 52 3 75 5 7 95 99 101 100 30 29 27 24 25 23 10 26 28 96 98 97 2 86 4 89 91 8 92 94 93 11 9 6 90 88 87 84 1 85 83 Route 4:0 114 115 112 110 107 108 105 104 65 67 103 20 21 22 106 109 111 113 117 116 42 122 123 132 129 127 124 43 126 121 125 143 147 145 83 Route 5:0 155 83	Route 1: 0 7 88 86 3 87 91 11 89 2 155 75 1 84 85 4 5 90 10 93 92 94 9 6 8 83 Route 2: 0 98 100 102 19 16 12 14 97 95 15 18 17 13 101 96 99 111 109 27 29 110 107 112 30 113 26 28 25 24 20 22 105 23 106 104 103 21 83 Route 3: 0 82 162 159 160 80 77 76 81 161 156 151 71 150 153 70 73 158 78 157 79 83 Route 4: 0 134 53 133 60 138 58 140 136 56 137 135 55 54 59 57 139 125 45 127 47 49 50 51 52 132 130 131 128 48 40 43 123 41 124 129 126 44 46 42 122 121 83 Route 5: 0 147 65 145 69 148 146 142 62 66 149 68 64 141 61 72 67 63 74 154 152 144 143 83 Route 6: 0 37 118 116 33 35 31 32 115 117 120 39 119 38 36 34 114 108 83
CMT12Y	Route 1:0 44 29 34 74 77 76 38 95 93 33 32 53 55 48 31 35 37 36 94 39 75 71 13 17 18 19 73 15 72 70 12 14 16 92 90 91 111 110 97 58 Route 2:0 23 22 25 51 103 100 42 96 41 40 46 101 45 105 49 102 99 43 47 104 87 88 27 24 26 28 30 107 52 50 106 89 56 108 109 112 57 54 98 85 82 84 86 59 58 Route 3:0 2 1 60 6 62 61 63 5 3 4 64 66 8 67 69 68 10 11 9 65 7 81 21 78 20 79 80 83 58	Route 1: 0 6 67 11 68 64 69 10 9 8 66 61 5 65 3 63 7 62 4 1 59 2 60 58 Route 2: 0 13 77 19 15 73 74 70 12 72 14 16 71 75 17 18 76 25 82 78 79 84 26 27 29 24 86 28 85 87 88 30 23 81 21 80 20 22 58 Route 3: 0 83 91 32 89 31 33 93 38 39 94 36 92 34 90 35 37 95 58 Route 4: 0 101 43 42 99 98 96 103 41 97 40 44 100 48 51 106 52 50 107 105 46 49 47 102 104 45 112 57 55 110 54 111 109 53 56 108 58
CMT13X	Route 1:0 63 113 111 110 51 105 104 101 31 99 44 42 102 103 100 45 98 48 97 47 46 49 50 106 109 108 107 52 53 54 57 95 94 93 92 36 96 59 34 35 90 61 62 64 112 60 56 55 58 30 24 88 85 22 27 28 32 91 33 65 Route 2:0 43 39 38 37 25 84 81 77 78 79 80 76 9 6 18 7 74 75 3 2 1 5 4 10 11 15 14 13 73 8 12 82 17 16 19 87 21 86 20 89 26 23 29 41 40 69 70 68 65 Route 3:0 83 67 72 71 66 65	Route 1: 0 66 1 2 71 76 75 73 8 13 12 77 79 80 15 14 78 9 74 72 11 10 70 7 6 68 69 67 5 3 4 65 Route 2: 0 18 83 81 17 82 30 22 16 86 85 26 23 21 20 92 37 96 108 109 59 52 107 53 56 55 60 58 62 61 111 113 63 112 64 110 54 57 65 Route 3: 0 25 84 19 87 89 32 35 28 24 88 34 91 33 31 27 29 90 36 51 106 103 99 39 42 38 94 97 93 41 44 101 50 105 95 43 40 45 98 100 48 47 46 102 49 104 65
CMT13Y	Route 1:0 66 130 41 39 58 40 32 42 55 56 60 131 43 129 128 127 48 44 31 34 46 45 126 125 52 53 54 57 59 124 123 122 47 118 121 120 117 29 119 49 50 51 116 132 65 115 113 61 112 114 62 64 63 133 146 145 75 138 139 147 107 83 Route 2:0 15 14 80 79 78 142 141 70 67 69 71 74 72 144 68 111 110 36 109 28 24 106 1 13 105 21 103 20 26 23 102 99 9 12 100 17 16 104 19 25 22 27 30 33 35 108 37 38 135 143 140 137 73 134 76 77 136 95 97 93 91 83 Route 3:0 149 82 148 81 101 18 96 11 4 88 89 6 5 84 85 2 86 3 87 10 7 90 92 94 98 8 83	Route 1: 0 53 66 133 130 63 131 60 59 126 57 125 54 127 61 132 64 62 128 65 129 52 58 55 56 113 40 118 124 50 114 122 44 120 47 46 119 49 117 41 123 51 45 43 116 83 Route 2: 0 149 82 84 15 98 94 11 13 8 95 91 12 14 97 4 10 90 92 86 88 6 9 7 93 89 5 3 87 1 85 2 96 18 101 81 148 83 Route 3: 0 139 145 135 146 79 78 80 147 142 72 140 144 73 76 75 74 141 71 138 67 134 69 70 137 136 143 68 77 83 Route 4: 0 100 22 109 16 17 99 103 108 21 104 102 107 23 20 19 106 24 25 105 26 28 34 29 36 35 27 32 31 30 33 42 115 48 121 111 112 38 39 37 110 83
CMT14X	Route 1:0 37 36 19 18 17 13 15 14 12 16 34 33 32 51 115 112 45 48 117 50 118 31 35 53 121 122 124 126 59 57 55 54 123 125 127 60 58 56 120 63 108 38 105 70 Route 2:0 26 25 28 107 101 49 47 43 40 64 69 66 62 61 68 41 42 44 52 29 27 30 102 103 104 86 89 106 39 46 109 131 129 128 135 110 111 114 116 119 113 70 Route 3:0 67 65 24 22 20 21 23 82 85 83 87 88 84 2 3 75 5 77 7 73 4 76 78 8 79 81 80 99 100 10 96 11 9 6 74 71 1 72 98 136 70 Route 4:0 97 94 95 93 92 91 90 133 132 134 130 70	Route 1: 0 48 114 49 52 50 119 115 45 51 118 117 116 47 113 46 109 41 40 112 44 42 110 111 43 108 70 Route 2: 0 83 13 85 87 16 86 14 15 18 17 19 89 88 84 82 12 11 81 9 6 74 4 80 10 8 78 76 79 71 1 2 75 3 7 77 5 73 70 Route 3: 0 56 60 58 125 123 127 120 53 54 124 121 55 122 126 57 59 135 131 62 129 64 68 128 61 133 136 69 66 67 134 63 132 65 130 70 Route 4: 0 90 21 93 24 91 20 22 92 23 94 99 26 96 25 95 28 100 29 97 27 98 30 72 70 Route 5: 0 103 33 32 106 34 105 31 101 102 35 104 36 39 107 37 38 70

CMT14Y	Route 1:0 44 55 34 76 79 78 38 97 95 33 32 53 59 48 31 35 37 36 96 39 77 73 13 17 18 19 75 15 74 72 12 14 16 94 92 104 108 93 115 113 60 Route 2:0 26 23 24 28 51 105 102 42 101 43 98 99 41 40 46 103 45 107 49 47 106 29 27 30 109 52 50 91 56 110 58 111 114 116 112 57 54 100 89 87 88 90 61 60 Route 3:0 85 84 86 11 70 10 69 68 8 4 63 65 5 67 7 3 64 66 62 1 2 6 9 71 83 82 81 80 20 21 22 25 60	Route 1: 0 73 17 78 18 13 77 79 19 16 74 76 14 12 72 15 75 68 71 6 60 Route 2: 0 98 43 104 47 101 46 41 40 42 99 102 44 103 93 38 32 33 35 37 97 91 95 31 92 34 96 36 94 39 60 Route 3: 0 66 69 70 10 8 11 9 4 64 63 3 7 67 65 5 1 61 2 62 85 27 87 26 86 23 22 82 83 88 30 90 28 89 24 25 29 84 20 81 80 21 60 Route 4: 0 100 49 106 109 51 108 50 107 52 48 105 45 59 55 111 58 115 110 53 113 56 54 57 116 112 114 60

Appendix C: Run three of VRPSPD

Instance	Initial Solution Routes	Final Solution Routes
CMT1X	Route 1:0 36 35 3 31 43 26 28 20 29 34 39 45 33 10 30 21 2 51 Route 2:0 11 4 40 13 24 23 22 7 25 41 42 15 37 17 44 19 12 51 Route 3:0 9 38 5 49 16 50 48 32 1 8 6 14 18 47 46 51 Route 4:0 27 51	Route 1: 0 23 35 20 36 3 31 28 26 7 8 22 1 48 43 24 6 51 Route 2: 0 27 32 38 50 16 21 34 9 49 5 11 51 Route 3: 0 4 18 14 25 13 41 19 40 42 12 47 51 Route 4: 0 2 29 30 10 39 33 45 15 44 17 37 46 51
CMT1Y	Route 1:0 40 41 17 42 33 10 34 36 3 22 28 35 29 21 30 39 45 19 13 18 51 Route 2:0 11 7 31 20 8 26 43 23 24 14 25 4 47 15 37 44 51 Route 3:0 49 50 2 32 27 6 48 1 16 9 38 5 12 46 51	Route 1: 0 18 14 6 43 7 26 31 28 3 36 35 20 22 1 51 Route 2: 0 46 5 12 47 4 51 Route 3: 0 32 11 9 16 21 2 29 34 50 30 39 49 38 51 Route 4: 0 27 8 48 23 24 25 13 19 40 41 45 10 33 15 37 44 42 17 51
CMT2X	Route 1:0 66 38 72 31 25 50 23 49 18 55 65 59 11 60 71 76 Route 2:0 53 19 20 15 37 36 1 61 42 41 64 69 70 54 14 76 Route 3:0 7 22 21 47 28 62 73 63 43 24 56 67 76 Route 4:0 3 32 39 58 10 9 44 16 33 74 48 5 76 Route 5:0 57 13 8 35 52 27 45 29 30 2 4 76 Route 6:0 26 34 46 12 40 17 51 6 68 76 Route 7:0 75 76	Route 1: 0 74 21 22 61 62 28 5 75 67 4 76 Route 2: 0 66 65 59 53 14 58 12 76 Route 3: 0 46 8 19 54 27 45 48 29 47 2 30 76 Route 4: 0 63 49 16 23 56 64 43 42 41 1 73 76 Route 5: 0 51 3 24 18 50 25 55 32 33 6 68 76 Route 6: 0 40 17 44 9 39 31 72 10 11 38 76 Route 7: 0 69 36 37 71 70 60 20 15 57 13 52 35 7 34 26 76
CMT2Y	Route 1:0 7 66 31 50 72 25 49 18 55 65 59 14 11 71 76 Route 2:0 60 20 37 36 61 42 56 23 41 64 69 70 67 26 76 Route 3:0 43 1 73 62 28 74 47 15 13 54 57 5 21 22 33 6 76 Route 4:0 44 9 10 58 53 38 39 32 3 63 24 76 Route 5:0 48 2 30 45 19 8 35 46 52 27 29 75 68 76 Route 6:0 16 51 17 12 40 34 4 76	Route 1: 0 6 33 73 63 23 56 49 3 44 16 17 51 76 Route 2: 0 66 59 11 65 10 58 39 72 38 26 12 76 Route 3: 0 34 13 27 4 46 35 14 19 54 57 8 53 7 76 Route 4: 0 31 9 32 24 18 50 55 25 40 76 Route 5: 0 75 68 2 43 42 1 41 64 22 61 62 74 28 76 Route 6: 0 52 67 76 Route 7: 0 45 30 29 15 47 69 48 21 60 71 70 36 37 20 5 76
CMT3X	Route 1:0 38 16 17 6 37 14 15 22 39 4 24 34 33 9 30 32 36 8 7 19 11 20 35 29 25 23 13 41 Route 2:0 10 31 1 3 12 28 26 40 21 2 5 18 27 41	Route 1: 0 27 31 11 19 7 36 8 18 38 14 37 16 17 5 6 13 22 15 40 2 41 Route 2: 0 21 23 4 39 25 24 3 29 33 35 34 9 20 30 1 10 32 12 26 28 41
CMT3Y	Route 1:0 49 36 46 17 37 38 45 8 47 11 20 30 51 33 50 9 24 34 35 32 19 7 48 59 Route 2:0 43 42 57 2 41 22 21 56 4 39 55 54 12 1 3 29 25 23 15 44 5 16 14 13 6 59 Route 3:0 10 31 52 18 40 53 58 26 28 27 59	Route 1: 0 53 28 12 1 33 50 35 34 29 51 9 20 32 30 10 31 52 27 59 Route 2: 0 2 42 57 43 38 16 44 14 15 37 5 18 8 17 45 48 47 46 36 49 11 19 7 59 Route 3: 0 13 6 58 40 21 22 41 23 4 56 55 25 39 54 24 3 26 59
CMT4X	Route 1:0 50 33 38 37 16 45 36 47 48 19 32 35 34 9 20 11 49 46 17 44 43 41 15 14 13 51 Route 2:0 10 30 1 31 8 5 23 25 29 3 12 24 4 21 26 28 40 2 22 39 42 6 18 7 27 51	Route 1: 0 1 9 33 35 34 24 29 3 20 32 30 50 27 28 51 Route 2: 0 18 45 5 17 46 8 48 47 49 36 19 11 10 7 31 51 Route 3: 0 6 40 4 25 39 23 22 41 14 43 38 44 37 16 42 15 13 2 21 26 12 51
CMT4Y	Route 1:0 37 49 47 46 36 64 63 65 34 24 25 29 51 50 33 9 35 66 20 30 32 11 19 48 17 38 71 Route 2:0 44 43 15 67 39 55 4 56 23 22 41 57 42 14 16 45 8 61 59 71 Route 3:0 28 68 12 3 54 21 26 40 58 2 13 6 60 52 7 10 31 69 1 70 62 18 5 71 Route 4:0 53 27 71	Route 1: 0 31 69 30 32 66 20 70 10 48 36 46 19 64 62 11 63 7 47 49 71 Route 2: 0 28 26 21 39 55 25 24 54 4 56 23 67 41 22 2 53 71 Route 3: 0 68 29 12 3 34 35 65 9 51 1 50 33 27 71 Route 4: 0 6 59 60 5 52 18 8 45 17 61 16 38 44 14 37 42 57 43 15 13 40 58 71
CMT5X	Route 1:0 38 14 37 42 15 43 44 16 61 45 36 64 33 35 49 11 32 66 71 65 63 19 47 46 17 59 41 74 Route 2:0 34 20 10 62 70 30 50 51 9 29 24 54 55 25 39 4 56 72 22 23 67 68 74 Route 3:0 27 57 73 21 26 28 12 3 40 53 58 2 13 6 60 52 18 5 48 31 69 1 7 8 74	Route 1: 0 45 8 48 47 36 46 17 61 60 18 52 26 12 28 74 Route 2: 0 53 73 72 4 23 39 25 29 68 1 70 20 50 33 34 3 24 55 54 74 Route 3: 0 7 19 32 10 31 62 63 49 64 11 66 65 71 35 9 51 30 69 27 74 Route 4: 0 40 58 21 67 56 22 41 2 57 15 14 43 42 16 44 38 5 37 13 59 6 74
CMT5Y	Route 1:0 81 38 14 37 42 15 43 44 16 17 46 47 19 63 33 35 11 49 64 36 45 61 85 78 50 86 Route 2:0 71 34 24 55 25 39 56 74 23 67 29 9 51 30 20 32 66 65 48 41 86 Route 3:0 84 8 7 10 31 70 62 82 83 18 60 5 6 59 57 2 75 4 80 76 68 54 72 21 73 22 28 86 Route 4:0 27 52 69 1 3 12 77 79 13 58 40 26 53 86	Route 1: 0 13 59 61 85 38 14 42 57 43 15 37 5 84 60 83 48 7 18 8 82 86 Route 2: 0 6 44 16 17 45 46 36 47 49 62 64 11 10 19 63 31 69 52 27 86 Route 3: 0 76 3 77 1 81 79 24 29 54 80 68 55 25 67 39 23 56 4 22 41 73 74 75 21 53 86 Route 4: 0 50 32 70 30 65 71 20 66 78 34 35 9 51 33 12 26 58 2 40 72 28 86

CMT6X	Route 1:0 36 35 3 31 43 26 28 20 29 34 39 45 33 10 30 21 2 51 Route 2:0 11 4 40 13 24 23 22 7 25 41 42 15 37 17 44 19 12 51 Route 3:0 9 38 5 49 16 50 48 32 1 8 6 14 18 47 46 51 Route 4:0 27 51	Route 1: 0 46 47 4 18 13 41 19 40 12 51 Route 2: 0 37 17 42 45 15 44 33 39 30 34 29 50 21 16 9 10 49 38 5 51 Route 3: 0 48 24 14 25 23 43 7 8 26 2 32 1 6 27 51 Route 4: 0 22 31 28 3 35 36 20 11 51
CMT6Y	Route 1:0 40 41 17 42 33 10 34 36 3 22 28 35 29 21 30 39 45 19 13 18 51 Route 2:0 11 7 31 20 8 26 43 23 24 14 25 4 47 15 37 44 51 Route 3:0 49 50 2 32 27 6 48 1 16 9 38 5 12 46 51	Route 1: 0 11 46 51 Route 2: 0 12 5 38 16 29 36 35 3 31 28 22 1 8 26 48 32 51 Route 3: 0 2 20 21 9 50 10 33 39 30 34 49 37 47 51 Route 4: 0 6 14 25 13 40 19 41 42 15 45 44 17 4 18 24 43 7 23 27 51
CMT7X	Route 1:0 66 38 72 31 25 50 23 49 18 55 65 59 11 60 71 76 Route 2:0 53 19 20 15 37 36 1 61 42 41 64 69 70 54 14 76 Route 3:0 7 22 21 47 28 62 73 63 43 24 56 67 76 Route 4:0 3 32 39 58 10 9 44 16 33 74 48 5 76 Route 5:0 57 13 8 35 52 27 45 29 30 2 4 76 Route 6:0 26 34 46 12 40 17 51 6 68 76 Route 7:0 75 76	Route 1: 0 68 76 Route 2: 0 7 8 54 57 15 13 46 4 76 Route 3: 0 27 52 29 5 48 36 37 47 21 45 30 75 76 Route 4: 0 20 71 70 60 69 61 22 28 2 62 74 73 76 Route 5: 0 53 19 35 67 26 76 Route 6: 0 6 33 63 64 43 1 42 41 16 49 23 56 76 Route 7: 0 32 44 9 39 72 38 65 10 31 58 76 Route 8: 0 51 40 25 50 55 18 24 3 12 17 76 Route 9: 0 34 66 59 14 11 76
CMT7Y	Route 1:0 7 66 31 50 72 25 49 18 55 65 59 14 11 71 76 Route 2:0 60 20 37 36 61 42 56 23 41 64 69 70 67 26 76 Route 3:0 43 1 73 62 28 74 47 15 13 54 57 5 21 22 33 6 76 Route 4:0 44 9 10 58 53 38 39 32 3 63 24 76 Route 5:0 48 2 30 45 19 8 35 46 52 27 29 75 68 76 Route 6:0 16 51 17 12 40 34 4 76	Route 1: 0 51 40 76 Route 2: 0 4 45 30 69 36 71 47 60 70 20 15 37 5 57 48 29 27 76 Route 3: 0 10 72 39 31 9 32 25 18 49 44 24 3 50 55 76 Route 4: 0 67 34 76 Route 5: 0 41 56 42 64 23 43 1 33 63 16 76 Route 6: 0 46 8 52 19 53 14 11 35 17 76 Route 7: 0 75 68 6 2 73 62 28 22 61 21 74 76 Route 8: 0 26 12 58 38 65 66 59 54 13 7 76
CMT8X	Route 1:0 38 16 17 44 6 37 43 15 23 25 29 35 20 11 7 19 36 32 9 33 34 24 4 39 22 41 42 14 13 45 Route 2:0 10 30 3 12 28 27 1 31 8 18 5 2 40 26 21 45	Route 1: 0 1 27 28 12 26 33 34 9 35 3 29 24 4 21 40 25 39 23 22 41 2 45 Route 2: 0 6 37 13 42 15 43 14 38 44 16 5 17 19 36 8 7 11 10 32 20 30 31 18 45
CMT8Y	Route 1:0 49 36 46 17 37 38 45 8 47 11 32 35 34 29 9 33 50 51 30 20 63 19 48 64 Route 2:0 44 43 15 23 25 24 55 54 4 39 56 21 22 41 2 57 42 14 16 61 60 5 59 64 Route 3:0 27 28 53 58 40 26 12 3 1 10 31 52 18 7 62 13 6 64	Route 1: 0 26 12 24 54 55 4 39 25 41 56 23 22 2 40 53 64 Route 2: 0 28 58 21 57 15 43 14 42 37 61 16 5 60 17 44 38 59 13 6 64 Route 3: 0 52 48 7 19 62 47 49 46 36 45 8 18 64 Route 4: 0 1 50 33 34 35 9 3 29 51 20 30 32 10 11 63 31 27 64
CMT9X	Route 1:0 50 33 38 37 16 45 36 47 48 19 32 35 34 9 20 11 49 46 17 44 43 41 15 14 13 55 Route 2:0 10 51 3 24 12 54 39 4 21 26 40 6 2 42 22 23 25 29 30 1 31 8 5 18 7 55 Route 3:0 52 27 28 53 55	Route 1: 0 53 40 4 21 22 41 39 25 23 54 29 24 34 33 50 1 27 55 Route 2: 0 52 7 18 6 5 8 45 17 16 37 44 42 38 14 43 2 15 13 55 Route 3: 0 49 47 46 36 48 19 11 32 10 31 20 30 51 35 9 3 12 28 26 55
CMT9Y	Route 1:0 49 36 46 17 37 38 45 8 47 11 20 30 51 33 50 9 24 34 35 32 19 7 48 62 Route 2:0 44 43 15 23 25 55 29 3 54 4 39 56 22 41 2 57 42 14 16 61 60 5 59 6 62 Route 3:0 27 28 12 26 21 40 53 58 13 18 52 10 31 1 62	Route 1: 0 27 28 31 1 50 30 34 33 35 9 51 20 32 10 11 19 48 46 47 49 36 7 52 62 Route 2: 0 5 61 16 38 44 14 42 59 17 8 45 60 18 13 62 Route 3: 0 26 21 40 53 58 2 39 23 4 25 55 3 24 12 29 54 56 22 41 15 43 57 37 6 62
CMT10X	Route 1:0 38 14 37 42 15 43 44 16 61 45 36 64 33 35 49 11 32 66 71 65 63 19 47 46 17 59 41 73 Route 2:0 34 20 10 62 70 30 50 51 9 29 24 54 55 25 39 4 56 72 22 23 67 68 73 Route 3:0 57 21 26 28 12 3 40 53 58 2 13 6 60 52 18 5 48 31 69 27 1 7 8 73	Route 1: 0 28 33 51 20 30 70 32 64 49 19 63 11 10 7 62 36 47 46 48 45 8 73 Route 2: 0 39 4 56 41 23 67 25 55 22 72 57 42 43 15 2 73 Route 3: 0 59 37 73 Route 4: 0 31 52 18 16 44 14 38 17 61 5 60 6 13 58 21 40 12 24 54 26 53 73 Route 5: 0 27 68 29 3 71 65 35 34 66 9 50 69 1 73
CMT10Y	Route 1:0 37 33 49 47 46 36 64 63 65 35 34 24 29 9 50 51 71 66 20 30 32 11 19 48 17 38 75 Route 2:0 44 43 15 67 25 55 4 39 56 23 22 74 41 57 42 14 16 45 61 59 6 75 Route 3:0 53 2 73 40 21 68 12 3 54 72 13 5 60 18 52 7 10 31 69 1 70 62 8 75 Route 4:0 58 26 28 27 75	Route 1: 0 45 8 46 47 48 36 64 49 11 63 32 20 66 35 65 71 51 9 69 75 Route 2: 0 1 10 30 31 70 50 34 29 24 55 68 3 33 12 54 26 75 Route 3: 0 28 21 72 56 23 67 39 25 4 41 22 73 74 43 15 2 42 57 58 53 40 75 Route 4: 0 13 16 14 38 44 37 17 61 6 59 5 60 52 18 7 62 19 27 75
CMT11X	Route 1:0 66 64 61 57 53 52 54 59 50 46 34 32 28 35 30 24 22 27 33 36 29 49 51 65 62 63 60 56 55 58 72 Route 2:0 48 44 42 39 38 37 25 26 21 20 23 16 9 12 17 19 31 41 47 45 43 40 72 Route 3:0 8 13 15 4 6 5 1 2 3 10 7 11 14 18 68 70 67 69 71 72	Route 1: 0 68 71 69 70 67 53 56 57 45 38 48 50 49 51 46 41 44 37 42 47 43 40 39 72 Route 2: 0 18 10 2 1 5 4 12 14 11 3 9 6 8 13 7 15 72 Route 3: 0 25 23 27 30 24 17 16 26 31 34 33 36 29 35 28 32 22 21 20 19 62 60 65 59 64 66 63 58 61 55 54 52 72
CMT11Y	Route 1:0 34 32 36 65 62 58 53 55 56 60 63 64 61 57 52 54 51 45 40 43 48 49 47 42 39 38 37 35 41 44 46 50 59 27 33 66 Route 2:0 30 24 22 16 17 19 25 23 21 20 26 28 29 31 12 14 11 3 2 1 5 4 10 9 15 13 8 66 Route 3:0 7 6 18 66	Route 1: 0 18 52 54 64 40 39 48 38 49 50 51 46 44 47 42 45 43 41 37 65 59 62 63 58 60 56 55 53 57 61 66 Route 2: 0 1 15 2 3 4 6 5 13 14 10 9 7 11 8 12 23 29 32 35 26 30 28 31 34 36 27 33 25 24 22 19 20 21 16 17 66

CMT12X	Route 1:0 70 71 76 82 81 78 77 80 79 73 63 45 56 60 58 53 83 Route 2:0 38 37 35 51 48 54 57 59 55 72 64 68 61 62 74 39 83 Route 3:0 47 31 32 34 19 36 33 50 49 52 44 42 41 40 46 83 Route 4:0 43 12 16 18 17 13 15 14 2 6 8 11 9 4 1 83 Route 5:0 30 29 24 27 28 26 10 3 75 5 7 23 21 20 22 25 66 65 67 69 83	Route 1: 0 82 71 70 73 78 76 80 79 77 81 63 83 Route 2: 0 72 74 64 61 62 69 68 66 65 67 83 Route 3: 0 27 25 24 26 29 34 32 30 28 21 22 23 20 83 Route 4: 0 58 60 53 56 57 55 59 54 83 Route 5: 0 33 35 31 36 39 37 38 52 49 48 45 50 47 42 51 46 44 41 43 40 83 Route 6: 0 18 15 14 16 12 19 17 13 9 11 10 7 4 2 1 75 5 6 8 3 83
CMT12Y	Route 1:0 14 12 19 18 38 35 45 47 33 37 36 39 17 13 15 16 58 Route 2:0 53 56 55 57 54 51 48 46 40 42 44 50 31 34 32 58 Route 3:0 30 29 52 41 43 49 24 27 28 26 11 10 8 6 7 5 3 4 1 2 9 23 21 20 22 25 58	Route 1: 0 20 25 32 33 35 36 37 31 38 39 34 29 30 28 26 27 24 22 23 21 58 Route 2: 0 41 45 46 49 47 50 52 42 43 40 44 51 48 54 57 53 56 55 58 Route 3: 0 18 17 15 16 12 14 19 13 1 2 4 6 9 8 10 7 5 3 11 58
CMT13X	Route 1:0 63 60 62 61 64 56 55 59 50 46 34 35 32 28 31 24 22 27 30 33 36 44 42 47 49 48 45 51 57 54 52 53 58 65 Route 2:0 43 39 38 37 25 16 8 9 6 13 12 17 19 26 21 20 23 29 41 40 65 Route 3:0 14 11 3 2 1 5 4 7 10 15 18 65	Route 1: 0 15 8 17 22 25 19 16 21 20 23 26 46 49 50 47 48 42 39 45 43 40 18 65 Route 2: 0 41 44 38 37 51 61 63 58 56 55 53 54 52 57 64 60 62 59 65 Route 3: 0 3 1 2 7 9 4 10 6 11 13 14 5 12 24 27 33 31 34 30 35 28 36 29 32 65
CMT13Y	Route 1:0 34 30 32 36 66 64 61 57 53 52 54 59 50 46 35 44 42 39 41 47 49 48 45 51 65 62 58 55 56 60 63 75 33 27 83 Route 2:0 80 79 78 72 70 69 71 74 73 76 68 43 37 31 24 22 16 14 17 19 25 23 21 20 26 28 29 38 40 77 83 Route 3:0 82 81 18 12 15 4 6 5 1 2 3 10 7 9 11 13 8 67 83	Route 1: 0 75 79 78 68 76 73 80 77 74 71 69 70 67 72 61 56 66 62 60 64 63 65 59 53 58 54 52 57 55 83 Route 2: 0 81 18 25 22 34 33 36 28 31 30 24 19 16 20 21 23 37 38 42 44 49 46 50 48 51 47 41 39 43 45 40 83 Route 3: 0 10 1 13 11 6 7 9 4 5 2 3 14 15 17 27 29 32 35 26 12 8 82 83
CMT14X	Route 1:0 53 55 45 59 57 54 58 60 56 64 63 38 70 Route 2:0 35 39 36 34 33 32 31 37 51 50 48 46 19 14 70 Route 3:0 52 12 15 13 17 18 16 2 6 8 10 11 9 4 1 70 Route 4:0 30 29 49 47 43 40 61 62 65 67 66 69 68 41 42 44 24 27 28 26 23 22 25 70 Route 5:0 3 5 7 21 20 70	Route 1: 0 58 60 56 53 54 55 57 59 62 65 67 66 69 64 68 61 63 70 Route 2: 0 34 38 37 35 33 36 39 28 26 30 29 27 24 22 21 23 70 Route 3: 0 20 25 32 31 46 43 49 52 50 51 47 44 45 48 41 42 40 70 Route 4: 0 13 18 15 19 17 16 14 12 11 4 8 10 9 6 5 3 7 1 2 70
CMT14Y	Route 1:0 14 12 19 18 38 35 45 47 33 37 36 39 17 13 15 16 60 Route 2:0 53 58 56 55 59 57 54 51 52 48 46 40 41 44 50 31 34 32 42 60 Route 3:0 30 29 49 43 24 27 28 26 11 10 8 6 7 5 3 4 1 2 9 23 21 20 22 25 60	Route 1: 0 21 22 20 27 26 23 24 25 29 30 28 32 38 39 36 37 35 34 33 31 60 Route 2: 0 2 5 7 3 9 11 13 17 18 15 14 12 19 16 8 6 4 1 10 60 Route 3: 0 44 46 45 51 48 52 49 50 47 43 40 41 42 57 55 59 56 58 53 54 60