VEHICLE ROUTING PROBLEM WITH VENDOR SELECTION, INTERMEDIATE PICK-UPS AND DELIVERIES

by
UĞUR EMEÇ

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

Sabancı University
Spring 2013

VEHICLE ROUTING PROBLEM WITH VENDOR SELECTION, INTERMEDIATE PICK-UPS AND DELIVERIES

APPROVED BY:

Assoc. Prof. Dr. Bülent Çatay ....
(Thesis Supervisor)

Assoc. Prof. Dr. Burçin Bozkaya ....
(Thesis Co-Supervisor)

Assoc. Prof. Dr. Kerem Bülbül ....

Assoc. Prof. Dr. Haluk Demirkan ....

Assist. Prof. Dr. Murat Kaya ....

DATE OF APPROVAL: ...15..07.2013

*To the only “**Reason**” of my humble existence…*

ACKNOWLEDGEMENTS

First of all, I want to express my sincere thanks and deepest gratitude to the only "**Reason**" of my humble existence and my family for their love, guidance, patience and support at all times throughout my graduate period. Without them, I can say without no doubt that it was not possible for me to complete this thesis.

I also want to thank to my advisors Dr. Bülent Çatay and Dr. Burçin Bozkaya for their invaluable guidance.

Besides those, I would like to express my thanks to TUBITAK for their valuable financial support that encouraged me to get my Master of Science degree. Their support was extremely essential for me to finalize my graduate studies.

Finally, thanks to all industrial engineering graduate students for their worthy friendship and kindly helps.

TEDARİKÇİ SEÇİMLİ, ARA DAĞITIM VE TOPLAMALI
ARAÇ ROTALAMA PROBLEMİ

Uğur Emeç
Endüstri Mühendisliği, Yüksek Lisans Tezi, 2013

Tez Danışmanı: Doç. Dr. Bülent Çatay
Doç. Dr. Burçin Bozkaya

Anahtar Kelimeler: Tedarikçi Seçimli Araç Rotalama, Ara Dağıtım ve Toplamalı Araç Rotalama, Premium Müşteri, ALNS Sezgisel Yöntemi, GIS

Özet

E-alışveriş, giderek artan hizmet yelpazesiyle birçok kişinin gündelik yaşamında günbegün daha vazgeçilmez olmaktadır. Bu çalışmada, çevrimiçi perakendecilerin envanter maliyetlerini arttırmadan, müşterilerinin ek gelir oluşturma ihtimali yüksek organik yiyecek, elektronik eşya, hediyeler vb. gibi özel ürün taleplerini karşılayabilecekleri dağıtım planlamasını yapmak için etkin bir model önerilmektedir. Önerilen model Tedarikçi Seçimli, Ara Dağıtım ve Toplamalı Araç Rotalama Problemi (TSADTARP) olarak nitelendirilmektedir. Söz konusu model, özel ürünlerin tedarik ağındaki uygun harici tedarikçilerden toplanarak müşterilere teslim edildiği bir dağıtım ağına dayanmaktadır. TSADTARP problemini çözmek için yeni ekleme, çıkarma ve tedarikçi seçme/atama mekanizmaları geliştirilerek bir Uyarlanabilir Geniş Komşuluklu Arama sezgisel algoritması önerilmektedir. Önerilen yaklaşımın performansı hem Solomon'un iyi bilinen zaman pencereli araç rotalama problemi örnekleri kullanılarak hem de bu probleme özgü yeni örnekler yaratılarak sınanmıştır. Yapılan kapsamlı analiz sonucunda önerilen sezgisel yöntemin makul sürede kaliteli çözüm elde etmede başarılı olduğu ortaya konulmuştur.

VEHICLE ROUTING PROBLEM WITH VENDOR SELECTION, INTERMEDIATE PICK-UPS AND DELIVERIES

Uğur Emeç
Industrial Engineering, Master's Thesis, 2013

Thesis Supervisor: Assoc. Prof. Dr. Bülent Çatay
Assoc. Prof. Dr. Burçin Bozkaya

Keywords: Vehicle Routing with Vendor Selection, Vehicle Routing with Intermediate Pickups and Deliveries, Premium Customer, ALNS Heuristic, GIS

Abstract

Online shopping is becoming nowadays more indispensable to many people in their daily lives with a growing service range for a wide variety of goods. In this thesis, we study a distribution planning model for online retailers to fulfill the diverse consumer demands especially for premium goods, i.e. goods with a high potential to create additional income such as organic food, electronic materials, special gifts etc., without increasing inventory related costs. We refer to the related distribution planning problem as the Vehicle Routing Problem with Vendor Selection, Intermediate Pick-ups and Deliveries (VRPVSIPD). The VRPVSIPD is based on a distribution network where premium goods are acquired from a proper set of external vendors at multiple locations in the supply network and delivered to customers. In order to solve the VRPVSIPD, we present an improved Adaptive Large Neighborhood Search (ALNS) heuristic by introducing new removal, insertion and vendor selection/allocation algorithms. To investigate the performance of the proposed methodology, we conduct an extensive computational study using both the well-known Solomon instances for Vehicle Routing Problem with Time Windows and newly generated benchmark instances for the VRPVSIPD. Our results reveal that the proposed methodology is effective in terms of both the solution quality and computational time.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

Chapter 1

Introduction

Shopping habits of consumers have rapidly changed especially in the last decade as a result of remarkable developments in e-commerce, with many more consumers who prefer online shopping in place of traditional in-store shopping for convenience. Consequently, leading and visionary online retailers look for up-and-coming business strategies to diversify their in-stock ("standard") products with outsourced ("premium") products, i.e. products with a high potential to create additional income, so as to satisfy and increase diverse consumer demand in a collaborative relationship with a set of external vendors. From this perspective, a popular online retailer AmazonFresh offers fresh grocery items like wine, pumpkin pie, vegetables, meat, seafood etc. for sale, as well as a subset of items from the main Amazon.com storefront. Another well-known online retailer Peapod provides grocers, vegetables, and many other by means of a centralized business model with two concepts, warehouse and ware rooms, where warerooms are the dedicated areas attached to a subsidiary international food provider Royal Ahold. Furthermore, originally UK-focused grocery retailer Tesco has diversified into areas such as the retailing of electronics, furniture, petrol, software, music downloads etc. through multiple subsidiary stores differentiated by size and the range of products/services provided. All these business settings are particularly based on vendor-managed inventories in which consumer demand for premium products is either fulfilled directly by external vendors or by online retailers where external vendors may store their products in fulfillment centers of online retailers. This thesis is aimed at developing an efficient solution procedure for the distribution planning problem that arises in these kinds of collaborative relationships by considering additional business rules to enable more online retailers to take advantage of these promising business strategies.

In this study, we consider a business model in which premium product orders are fulfilled directly from online retailers without requiring external vendors to store their products in fulfillment centers. Within this context, the proposed model is built on a distribution network which consists of (1) depot, i.e. a store of an online retailer that only supplies "standard" products offered by the retailer, (2) vendors, i.e. external stores that are only eligible to supply their individual "premium" products, (3) "regular" customers, i.e. customers that only purchase standard products, and (4) "premium" customers, i.e. customers that *additionally/only* purchase premium products. The routing of the regular customers only is straightforward with respect to classical considerations in the Vehicle Routing Problem (VRP) setting, whereas the routing of premium customers present an additional challenge as two simultaneous decisions are made: i) allocation of vendor(s) to each premium customer so as to satisfy his/her entire premium product order, ii) routing of regular customers and premium customers along with their respective vendor set while preserving feasibility concerns such as precedence, vehicle capacity, time windows, etc. Under these circumstances, the delivery of the goods to each premium customer takes place only after the entire set of premium products are collected and are combined with the standard products already loaded at the depot. We refer to this problem as Vehicle Routing Problem with Vendor Selection, Intermediate Pickups and Deliveries (VRPVSIPD).

The contributions of this study can be summarized as follows:

- A mathematical model formulated to minimize total transportation costs subject to additional business rules.
- An improved Adaptive Large Neighborhood Search (ALNS) procedure is developed to solve the VRPVSIPD. The proposed ALNS improves some of the existing removal/insertion mechanisms and introduces new removal/insertion and vendor selection/allocation mechanisms specific to the VRPVSIPD. It also presents additional scoring mechanisms for self-adjustment of some removal/insertion procedures and introduces multiple initial solutions during the search.
- The best-known solutions for the five real numbered VRP with time windows (VRPTW) instances of Solomon (1987) are achieved. Furthermore, the upper bound of the only open truncated instance, i.e. R208, is improved.

- New data for VRPVSIPD are randomly generated using Solomon's VRPTW instances and the results are reported as benchmarks for future studies.
- Finally, a case study based on a real dataset in the city of Istanbul, Turkey is presented. The proposed ALNS heuristic is integrated with the ArcGIS environment to provide a convenient user interface and to present analysis results including total route distance, duration, and vehicle utilization under two different scenarios.

The remainder of this thesis is organized as follows. Chapter 2 reviews the related literature. Chapter 3 describes VRPVSIPD and presents the mixed integer linear programming model. The improved ALNS algorithm is detailed in Chapter 4. Computational results are given in Chapter 5, which is followed by a case study in Chapter 6. Finally, Chapter 7 concludes the thesis with some remarks on future research directions.

Chapter 2

Literature Review

In VRPVSIPD, the vehicles start their trips with a load of standard products available at the depot and each customer is visited only once by a single vehicle that delivers its entire basket of orders. Accordingly, a proper set of vendor pick-up locations is matched to each premium customer and the corresponding vehicle stops at these vendor locations to collect ordered premium products before the delivery takes place. In this context, transfers are not allowed between vehicles and the minimum distance solution is sought in the presence of precedence, time windows, and capacity constraints. The importance of using multiple sourcing and consolidation points to fulfill premium orders of consumers in e-grocery environment is first identified by Bozkaya et al. (2009). In a succeeding study Yanik et al. (2013) investigate the role of premium product offerings in creating critical mass and profit, and propose a hybrid metaheuristic approach using a genetic algorithm for vendor selection-allocation phase followed by a modified savings algorithm for the vehicle routing phase. The proposed genetic algorithm guides the search for optimal vendor pick-up location decisions. The authors also show possible profit opportunities of the new business model on a case study using a real dataset.

Other methods that are closest to the VRPVSIPD can be generally categorized into VRP with intermediate facilities (VRP-IF) and VRP with satellite facilities (VRP-SF). Angelelli and Speranza (2002) study periodic VRP in which some intermediate facilities exist where the vehicles renew their capacity for a collection problem. They propose a tabu search algorithm and present computational results on a set of randomly generated instances. Sevilla and de Blas (2003) take into account time windows in VRP with intermediate facilities and propose an algorithm that is based on neural networks and an ant colony system. Tarantilis et al. (2008) address the case where vehicles start their trips from a central depot and the intermediate depots act as replenishment stations. In order to create inter-depot routes, they assume all vehicles are centralized in a single depot. They propose a three-step algorithmic framework for solving their problem. An initial solution is first obtained by a cost-saving construction heuristic. Then, this solution is improved by employing tabu search within the variable neighborhood search methodology. Finally, a guided local search is applied to eliminate low-quality features from the final solution produced. Polacek et al. (2008) develop a simple and robust variable neighborhood search algorithm to solve the capacitated arc routing problem with intermediate facilities. Liu et al. (2010) consider waste collection problems in the presence of intermediate facilities where the vehicles are unloaded when they are full. They develop an improved ant colony system algorithm for the Multi-Depot Vehicle Routing Problem with Inter-Depot Routes. Bard et al. (1998a, 1998b) consider satellite facilities to replenish vehicles during a route. They present a branch-and-cut methodology for solving the VRP-SF problems subject to capacity and route time constraints. In all these cases, intermediate/satellite facilities are considered to be identical and it is enough to stop by just one of them for replenishment.

If all additional business rules are left aside, then the VRPVSIPD problem can be considered as a variant of VRP with pick-ups and deliveries (VRPPD) in which goods are transported from pickup to delivery points where all pickups must be made before the deliveries. Since the VRPPD has been reported as NP-hard in Toth and Vigo (2002a, 2002b), being a generalization of the capacitated VRP, the VRPVSIPD problem is also NP-hard. Many exact and heuristic methods for the VRPPD have been proposed so far, but it is out of the scope of this study to give an overview of all these methods. Instead we refer the interested reader to Desaulniers et al. (2002), Cordeau and Laporte (2003a, 2007), Nagy and Salhi (2005), Gendreau et al. (2007), Berbeglia et al. (2007, 2010), Cordeau et al. (2008), and Parragh et al. (2008a, 2008b) for extensive reviews of the problem classifications, formulations, exact, and metaheuristic approaches proposed for solving VRPPD and its variants.

Jaw et al. (1986) propose insertion procedures, whereas Cullen et al. (1981), Bodin and Sexton (1986), Dumas et al. (1989), Desrosiers et al. (1991), Toth and Vigo (1996), and Borndörfer et al. (1997) consider cluster-first and route-second methods to deal with the VRPPD. Furthermore, Toth and Vigo (1997), Nanry and Barnes (2000), Caricato et al. (2003), Cordeau and Laporte (2003b), Attanasio et al. (2004), Montane and Galvao (2006), and Melachrinoudis et al. (2007) use tabu search to solve the VRPPD. Rekiek et al. (2006) and Ganesh and Narendran (2007) make use of genetic algorithms, whereas Hart (1996) and Li and Lim (2001) consider simulated annealing approach for the VRPPD. Doerner et al. (2001, 2003) propose ant colony optimization and Bent and van Hentenryck(2006) develop a hybrid heuristic for the VRPPD with time windows. Lin (2008) compares a classical VRPPD model to a model where vehicle fleet is split into pickup vehicle fleet and delivery vehicle fleet, and transfers take place between these fleets at the depot. Thangiah et al. (2007) introduce transfer points in the network different from the depot and allow transfers between the vehicles. They assume a divisible delivery to the customer, which leads to multiple numbers of vehicles delivering goods to the customer.

In terms of accuracy as well as flexibility, the approach introduced by Ropke and Pisinger (2006a) is one of the best methods for the VRPPD at hand. They propose a metaheuristic denoted by Adaptive Large Neighborhood Search (ALNS) as an extension of the Large Neighborhood Search (LNS) framework put forward by Shaw (1998). This metaheuristic aims to improve an initial feasible solution progressively by means of multiple removal and insertion algorithms competing in an adaptive environment to diversify and intensify the search. It consists of two main phases generally named as *destroy* and *repair*. At each iteration, one destroy algorithm and one repair algorithm are selected independently based on their historical performance to modify and reconstruct the current solution. The resulting new solution is accepted if the predefined local search framework (e.g. simulated annealing, tabu search) criteria are met. Ropke and Pisinger (2006b) develop a unified ALNS heuristic for a large class of vehicle routing problems with backhauls. They further improve this heuristic with additional algorithms in Pisinger and Ropke (2007) and show that the ALNS framework produces good results for different types of VRP's.

The ALNS framework can be successfully applied to a wide range of problems as a result of its flexibility. Muller (2009) presents an ALNS heuristic for the resource-constrained project scheduling problem. Cordeau et al. (2010) schedule technicians and tasks in a large telecommunications company by means of an ALNS framework. Furthermore, Muller et al. (2012) propose a hybrid ALNS heuristic for lot-sizing problem with setup times. Masson et al. (2012) propose an ALNS heuristic for a variant of the pick-up and delivery problem where requests can be transferred between vehicles during their trip at specific locations called transfer points. They introduce new destruction and repairing heuristics dedicated to the use of these transfer points. Ribeiro and Laporte (2012) develop an ALNS heuristic for a variant of the classical capacitated vehicle routing problem with an objective to minimize the sum of arrival times at customers, instead of the total routing cost. Demir et al. (2012) present an improved ALNS heuristic for Pollution-Routing Problem (PRP) by introducing some novel removal and insertion heuristics with promising results. Kristiansen et al. (2013) solve the consultation timetabling problem at Danish high schools with an ALNS framework. We refer the further interested reader to Pisinger and Ropke (2010) for a recent survey on large neighborhood search, its variants and extensions like the ALNS framework.

In this study, the proposed ALNS heuristic modifies and improves some of the existing removal and insertion procedures in the literature. Also, it introduces new removal/insertion and vendor selection/allocation procedures specific to the VRPVSIPD problem. Furthermore, additional adaptive scoring mechanisms for self-adjustment of some removal/insertion procedures and multiple initial solutions during the search are introduced.

Chapter 3

Problem Description and Formulation

In this chapter, the description of VRPVSIPD is presented along with a mixed integer linear programming model which can be used to obtain optimal distribution plans for small size instances.

3.1 Problem Description

Let $G = (N, A)$ be an undirected complete network and $\{0\} \in N$ denote the depot. $F = N \backslash \{0\}$ is partitioned into a subset V of external vendors and a subset $C = C^R \cup C^P$ of regular as well as premium customers, respectively. In this distribution network, the depot only provides standard in-store products whereas external vendors are not required to be identical and only provide a set of premium products denoted by P. Standard product demand, which is also the consumption of vehicle capacity, of each customer $c \in C$ is denoted by d_c. Premium product supply range of external vendors is represented with a binary matrix denoted by $A = \left[a_{vp} \in \{0,1\}\right]_{|V| \times |P|}$, i.e. $a_{vp} = 1$ if vendor $v \in V$ supplies premium product $p \in P$. Demand indicator of premium customers is also represented with a binary matrix denoted by $B = \left[b_{cp} \in \{0,1\}\right]_{|C^P| \times |P|}$, i.e. $b_{cp} = 1$ if premium customer $c \in C^p$ requests a premium product $p \in P$. Furthermore, premium orders of premium customers are represented with a matrix denoted by $Q = \left[q_{cp} \in \mathbb{Z}\right]_{|C^P| \times |P|}$. Note that B and Q are defined separately to make integer programming process more perceptible. Each premium product $p \in P$ is associated with a volume of w_P and premium product demand of each premium customer $c \in C^p$ is derived from $\sum_{p \in P} q_{cp} w_p$.

In this setting, it is assumed that the depot and external vendors have unlimited supplying capacities. Also, premium order $p \in P$ of premium customer $c \in C^p$ can be supplied by just one of the proper external vendors. Additionally, customers are served by a fleet K which consists of homogenous vehicles with capacity $\mathcal{U}$. The vehicles are located at the depot and some may stay idle if there is not sufficient demand. Each vehicle $k \in K$ is allowed to leave the depot just for once, i.e. vehicles can be assigned to utmost one route. Furthermore, a multiple pick-up and single delivery principle is assumed for each customer $c \in C$. That is, a premium customer $c \in C^p$ can only be served after his entire premium orders have been collected from the appropriate set of external vendors and combined with his standard products in the same vehicle. Each node $i \in F$ is associated with a service time of s_i units, which represents loading time for external vendors, unloading time for customers, and is assumed to be negligible for the depot. Travel time from node $i \in N$ to node $j \in N$ is denoted as t_{ij}, and the associated cost of this travel is defined as c_{ij}. Each node $i \in N$ has a beginning and end time window for service represented with beg_i and end_i respectively. In this context, deliveries to customers are limited to working hours of the depot as well as vendors.

A simple problem setting for the VRPVSIPD and the corresponding optimal solution is given in Figures 3.1a and 3.1b, respectively. In Figure 3.1a, (x, y) pair adjacent to each node represents the *X* and *Y* coordinates in the Euclidean space, where c_{ij} and t_{ij} values are rounded down to one decimal. The depot is represented with node D. There are four vendors where $V = \{\text{V1}, \text{V2}, \text{V3}, \text{V4}\}$, six regular customers where $C^R = \{\text{C1}, \dots, \text{C6}\}$, three premium customers where $C^P = \{\text{C7}, \text{C8}, \text{C9}\}$, and three premium products where $P = \{1,2,3\}$, with corresponding volume of $w_{\{1,2,3\}} = \{5,4,6\}$. Standard product demand of customers is $d_{\{\text{C1},\dots,\text{C9}\}} = \{10,30,20,10,40,20,5,10,8\}$ and premium product supply range of external vendors is $A = \big[[1,0,0],[0,1,0],[0,1,0],[0,0,1]\big]$. Premium demand indicator matrix and premium order matrix of premium customers are $B = \big[[1,0,1],[1,1,0],[0,0,1]\big]$ and $Q = \big[[3,0,2],[1,5,0],[0,0,10]\big]$, respectively. There are two available vehicles, i.e. $K = \{1,2\}$, each with a capacity of $\mathcal{U} = 100$. Service time of each node is negligible, i.e. $s_i = 0$ for each $i \in F$. Beginning and end time windows of each node $i \in N$ are $beg_i = 0$ and $end_i = 200$, respectively.

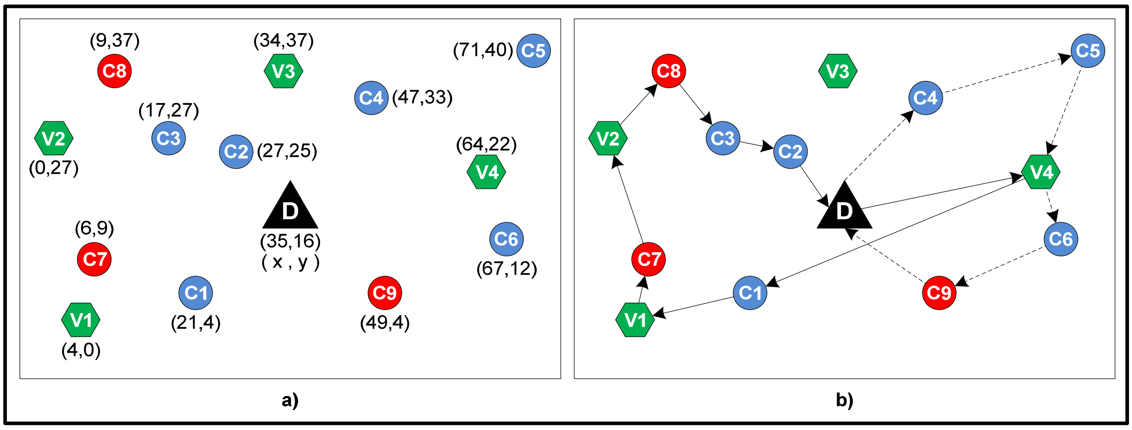


Figure 3.1: A simple illustration for the VRPVSIPD problem. **a)** Problem setting **b)** Optimal solution with corresponding routes.

The optimal solution of the problem setting given in Figure 3.1a is illustrated in Figure 3.1b with a total distribution cost(z) of 283,5. There are two routes as $Route_1$: D → V4 → C1 → V1 → C7 → V2 → C8 → C3 → C2 → D with a total distance of 170 and $Route_2$: D → C4 → C5 → V4 → C6 → C9 → D with a total distance of 113,5.

3.2 Mixed Integer Linear Programming Formulation

The proposed mathematical model includes binary as well as continuous decision variables. If vehicle $k \in K$ travels from node $i \in N$ to node $j_{\{j \neq i\}} \in N$, then binary variable x_{ijk} takes value one and value zero otherwise. In addition, if premium customer $c \in C^P$ is served by vendor $v \in V$ with vehicle $k \in K$, then binary variable z_{cvk} is equal to one and zero otherwise. Furthermore, if premium order $p \in P$ of premium customer $c \in C^P$ is supplied by vendor $v \in V$ with vehicle $k \in K$, then binary variable r_{cvpk} takes value one and zero otherwise. Continuous variable T_{ik} represents arrival time of vehicle $k \in K$ at node $i \in N$. Moreover, continuous variable L_{ik} stands for load of vehicle $k \in K$ when it leaves node $i \in N$. Then, the MILP formulation of VRPVSIPD can be given as follows:

Minimize $\quad z = \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ijk}$ (1)

Subject to:

$\sum_{j \in F} x_{0jk} \leq 1 \;, \; \forall k \in K$ (2)

$\sum_{k \in K} \sum_{j \in N} x_{jck} = 1 \;, \; \forall c \in C$ (3)

$$\sum_{j\in N} x_{ijk} = \sum_{j\in N} x_{jik} \ , \ \forall i \in N,\ k \in K \tag{4}$$

$$M \sum_{j\in N} x_{jvk} \ \geq \ \sum_{c\in C^P} z_{cvk} \ , \ \forall v \in V,\ k \in K \tag{5}$$

$$M \sum_{j\in N} x_{jck} \ \geq \ \sum_{v\in V} z_{cvk} \ , \ \forall c \in C^P,\ k \in K \tag{6}$$

$$T_{vk} + s_v + t_{vc} \leq T_{ck} + M\,(1 - z_{cvk})\,,\ \forall k \in K, v \in V, c \in C^P \tag{7}$$

$$T_{ik} + s_i + t_{ij} \leq T_{jk} + M\left(1 - x_{ijk}\right),\ \forall k \in K, i \in N, j \in F \tag{8}$$

$$beg_i \leq T_{ik} \leq end_i, \forall i \in N, k \in K \tag{9}$$

$$T_{ik} + x_{i0k}(t_{i0} + s_i) \leq M(1 - x_{i0k}) + end_0, \forall i \in N, k \in K \tag{10}$$

$$z_{cvk} \geq r_{cvpk} \ , \ \forall c \in C^P, v \in V, p \in P, k \in K \tag{11}$$

$$\sum_{k\in K} \sum_{v\in V} a_{vp} r_{cvpk} \ = \ b_{cp} \ , \ \forall c \in C^P, p \in P \tag{12}$$

$$L_{0k} \ = \ \sum_{c\in C} \sum_{j\in N} d_c x_{jck} \ , \ \forall k \in K \tag{13}$$

$$L_{jk} - (d_c + \sum_{p\in P} q_{cp} w_p) \leq L_{ck} + M\left(1 - x_{jck}\right),\ \forall c \in C^P, j \in N, k \in K \tag{14}$$

$$L_{jk} - d_c \ \leq \ L_{ck} + M\left(1 - x_{jck}\right),\ \forall c \in C^R, j \in N, k \in K \tag{15}$$

$$L_{jk} + \sum_{c\in C^P} \sum_{p\in P} r_{cvpk} q_{cp} w_p \ \leq \ L_{vk} + M\left(1 - x_{jvk}\right),\ \forall v \in V, j \in N, k \in K \tag{16}$$

$$L_{ik} \ \leq \ \mathcal{U} \sum_{j\in N} x_{ijk} \ , \ \forall i \in N, k \in K \tag{17}$$

$$x_{ijk} \in \{0,1\}\,, \forall i,j \in N, i \neq j, k \in K \tag{18}$$

$$y_k \in \{0,1\}\,, \forall k \in K \tag{19}$$

$$z_{cvk} \in \{0,1\}\,, \forall c \in C^P, v \in V, k \in K \tag{20}$$

$$r_{cvpk} \in \{0,1\}\,, \forall c \in C^P, v \in V, p \in P, k \in K \tag{21}$$

The objective function z is given by (1) and aims to minimize total distribution costs associated with the distance travelled. Constraint (2) ensures that a vehicle starts its route from the depot. Also, Constraint (2) along with Constraint (8), which is essential for the subtour elimination process, assures that a vehicle cannot appear in any one of the routes if it is not used. Notice that, if a vehicle does not start its route from the depot, then it cannot take place in any one of the routes as a result of the subtour elimination constraint (8). Constraint (3) is needed to assure that each customer is visited once, as vehicle flow balance is guaranteed by means of Constraint (4). Additionally, Constraint (4) together with the definition of decision variable x_{ijk}, Constraint (2) and Constraint (8) ensures that each route ends at the depot. Furthermore, the coordination of the decision variables in order to form consistent vendor as well as premium customer assignments and schedules throughout each route is provided by means of Constraint (5) and Constraint (6). Premium product orders of premium customers are collected from appropriate set of vendors before the time of delivery as a result of Constraint (7). Moreover, time consistency between consecutive stops in each route is ensured via Constraint (8). Constraint (9) guarantees that each customer is served within a prespecified time window, whereas Constraint (10) ensures that all vehicles return to the depot within the corresponding time window. Additionally, Constraints (8) and (9) make sure that routes do not contain any subtours. Constraint (11) is used to provide consistency between the decision variables related to vendor, premium customer, and premium product assignments so that a routing solution is formed properly throughout the model. The fact that different types of premium product orders of customers are supplied by exactly one of the proper vendors is ensured with Constraint (12). Constraint (13) assures that the initial load of each vehicle equals to the total standard product consumption of all customers who are assigned to that vehicle. Constraints (14) and (15) ensure that the load balance of each vehicle is consistent throughout its route after each delivery to premium customers and regular customers assigned to that vehicle, respectively. Similarly, Constraint (16) provides that the total amount of premium product load to be collected at each vendor is coherent with the formation of each route. On the other hand, capacity violation of each vehicle is prevented throughout the network by Constraint (17), whereas Constraints (18), (19), (20), and (21) are regular binary constraints for decision variables.

VRPVSIPD model given in this study differs from the models presented in Bozkaya et al. (2009) and Yanik et al. (2013). First of all, we only focus on distance minimization whereas the others take into account vehicle minimization as well. Moreover, we introduce new mechanisms to dynamically follow which premium customer gets service from which vendor for which premium product. These mechanisms let us consider more realistic scenarios in which "*super vendors*", i.e. vendors that supply more than one premium product at the same time, are used during the generation of the distribution plans. Furthermore, we improve the premium product handling process by separating order and individual consumption amounts of each premium product. By this way, we aim to easily adopt the proposed model to a future research scenario in which a premium customer may get service from multiple vendors with multiple vehicles for the same premium order.

Chapter 4

Solution Methodology

In this chapter, we propose an improved Adaptive Large Neighborhood Search (ALNS) framework for the VRPVSIPD problem.

4.1 Adaptive Large Neighborhood Search

The ALNS approach proposed in this study consists of three main sets of algorithms: (i) A_R: Removal, (ii) A_V: Vendor Selection/Allocation, and (iii) A_I: Insertion algorithms. It combines the strengths of the ALNS heuristics previously put forward by Ropke and Pisinger (2006a, 2006b), Pisinger and Ropke (2007), and Demir et al. (2012) by modifying some of the existing removal and insertion algorithms as well as introducing new removal, insertion and vendor selection/allocation algorithms specific to VRPVSIPD. It also introduces additional scoring mechanisms for self-adjustment of some removal/insertion mechanisms and produces multiple initial solutions during the search. The main aspects of our proposed ALNS heuristic for VRPVSIPD are as follows:

1. **General Flow:** Let S_C be a current feasible solution on hand at the beginning of each iteration and S_P be a partial feasible solution. At each iteration, a removal algorithm $r_a \in A_R$, a vendor selection/allocation algorithm $v_a \in A_V$, and an insertion algorithm $i_a \in A_I$ are dynamically and independently selected. Next, S_P is obtained by removing n_c regular and premium customers from S_C by using the selected removal algorithm r_a. Then, a new proper vendor set is selected and

appended to each removed premium customer via the selected vendor selection/allocation algorithm v_a. Finally, each removed customer is inserted into S_P with its respective vendor set, if any, by means of the selected insertion algorithm i_a. At the end of the iteration, we obtain a temporary feasible solution S_T which can be discarded or made the new S_C.

2. **Large Neighborhood:** The neighborhood size is determined by n_c and it has a substantial effect on the performance of the ALNS heuristic. On the one hand, if $n_c \ll |C|$ the effect of a large neighborhood is lost and therefore the search space may not be explored in an efficient way. On the other hand, if n_c is significantly large with respect to $|C|$ repairing S_P may be very time-consuming and/or the resulting S_T may be in poor quality (Pisinger and Ropke, 2010).

3. **Adaptive Scoring:** Let the entire search last N_I iterations and be divided into Δ *segments*, i.e. number of sequential iterations in size of $N_{SI} = \lfloor N_I/\Delta \rfloor$. Also, let π_a be the *score* of algorithm a (e.g. $a \in A_R$, $a \in A_I$, or $a \in A_V$ for removal, insertion and vendor selection/allocation algorithms, respectively) that represents how well algorithm a has performed during a *segment*. At the beginning of each *segment*, previous π_a values of all algorithms are set to zero. If a new global best solution S_B, i.e. the best solution found during the search, is found in an iteration of a *segment*, then π_a scores of r_a, v_a, and i_a algorithms are increased by σ_1 regardless of which one of them has yielded this improvement. If S_T is better than S_C, then π_a scores of r_a, v_a, and i_a algorithms are increased by σ_2. On the other hand, if S_T is accepted even though it is worse than S_C, then π_a scores of r_a, v_a, and i_a algorithms are increased by σ_3. Note that we assume $\sigma_1, \sigma_2, \sigma_3 > 0$.

4. **Adaptive Weight Adjustment:** Let $w_{a,s}$ and $\tau_{a,s}$ represent the adaptive weight of algorithm a (e.g. $a \in A_R$) and the number of times algorithm a has been selected during *segments* $= 1, \dots, \Delta$, respectively. If $s = 1$, then $w_{a,1} = 1$, i.e. initially all algorithms have the same weight. After N_{SI} iterations, i.e. at the beginning of *segment* $s + 1$, $w_{a,s+1}$ value is updated for each algorithm a according to the π_a scores obtained during the previous *segments* as follows:

$$w_{a,s+1} = \begin{cases} w_{a,s} & ,\tau_{a,s} = 0 \\ (1-\rho)w_{a,s} + \rho\,\pi_a/\tau_{a,s} & ,\tau_{a,s} > 0 \end{cases} \quad (22)$$

where $\rho \in [0,1]$ is a parameter called as *reaction factor* that controls how quickly the adaptive weight adjustment mechanism reacts to changes in the effectiveness of the algorithms.

5. **Adaptive Selection:** r_a, v_a, and i_a algorithms are independently and individually selected by means of a roulette wheel mechanism based on their past performances which are dynamically updated with respect to their adaptive weights using (22). Given m algorithms with $l = 1, \dots, m$ (e.g. $m = |A_R|$), let P_a^s be the probability of selecting algorithm a (e.g. $a \in A_R$) during $segments$. Then P_a^s is given by:

$$P_a^s = w_{a,s}/\sum_{l=1}^{m} w_{l,s} \quad (23)$$

6. **Initial Solutions:** The entire search starts with an initial feasible solution S_I obtained using the sequential *Greedy Heuristic* or *Regret-2 Heuristic* described in Section 4.1.3. The selection of one of these heuristics is controlled by a roulette-wheel mechanism similar to (23). At the beginning of the search, these constructive heuristics have equal selection chances and their *scores* are set to zero. Throughout N_I iterations, their adaptive weights are readjusted using (22) in which π_a values are only updated by σ_1 and σ_2. If there is no improvement in S_B after N_{IwI} (number of iterations without improvement) iterations, then a new S_I is introduced to diversify the solution space and to make better use of previously trained removal, insertion as well as vendor selection/allocation algorithms in the remaining search.

7. **Acceptance and Stopping Criteria:** Simulated Annealing (SA) local search framework is used at the master level of our proposed ALNS heuristic. Let $z(S)$ be the cost of a feasible solution S given by (1) and $T > 0$ be the current *temperature* which is initially set to T_{Start}. Then, S_T is always accepted if $z(S_T) < z(S_C)$, otherwise, it is accepted with probability $exp\,(-(z(S_T) - z(S_C))/T)$. Similar to Ropke and Pisinger (2006a), T_{Start} is set in such a way that S_T is accepted with probability *0.5* if it is μ (*start temperature control parameter*)

percent worse than S_C. Also, the current temperature T is gradually decreased every iteration using the expression $T = T\varepsilon$, where $0 < \varepsilon < 1$ is a fixed parameter named as *cooling rate*. The proposed ALNS heuristic stops after N_I iterations and returns S_B.

8. **Applying Noise:** Since some insertion heuristics generally tend to make the move that seems best locally and get stuck in a local optimum, a noise term may be added to the objective function in order to avoid this myopic behavior (2006a). Therefore, some of the insertion heuristics in this study are split into two sub categories: (i) *Clean*, which considers the $actual\ cost$, i.e. additional increase in the objective value as a result of a node insertion and (ii) *Noise Imposed*, which considers a $noised\ cost = actual\ cost + max_{i,j \in N}\{c_{ij}\}\ \ni\ \partial$ where $\ni$ is a noise parameter used for diversification and $\partial \in [-1,1]$ is a random number. The selection of Clean or Noise Imposed subcategory is determined using a roulette wheel procedure similar to (23) after an insertion heuristic has been adaptively selected. Note that performance of Clean and Noise Imposed insertion heuristics are tracked at a general level depending on how well each individual insertion heuristic performs with and without noise.

In the following sections, thirteen removal, six vendor selection/allocation and four insertion algorithms are described respectively.

4.1.1 Removal Algorithms

The destroy mechanism of the proposed ALNS framework uses one of the removal algorithms $r_a \in A_R$ given in this section to derive S_P by removing n_c customers from S_C and adding them into a removal list $\mathcal{L}$. A pseudo-code of the generic removal procedure is presented in Algorithm 4.1. The parameter θ represents the number of removal iterations, in each of which a subset of customers is removed from S_C by means of r_a and the routes are updated accordingly. The subsequent removal iterations consider the last updated routes to determine the next subset of customers to be removed. This procedure continues until n_c customers are removed. A regular customer $c_r \in C^R$ can be removed from its respective route R_{c_r} without any feasibility concerns. On the other hand, if a premium customer $c_p \in C^P$ is removed from its respective route R_{c_p}, then each external vendor in its current vendor set $V_{c_p} \subseteq V$, i.e. the set of vendors responsible to serve the premium customer c_p, is checked whether or not it serves any other premium customers in R_{c_p}. Only under the latter condition, the corresponding external vendor $v \in V_{c_p}$ is also removed from R_{c_p} of S_C, but it is not added into $\mathcal{L}$. Notice that an external vendor $v \in V_{c_p}$ is not removed from its respective route in S_C to maintain service feasibility if the former condition holds.

Algorithm 4.1: The generic structure of the removal procedure

Input: $r_a \in A_R, S_C, and\ maximal\ number\ of\ iterations\ \theta$

$Initialize\ \mathcal{L}\ (\mathcal{L} \longleftarrow \emptyset)$
$\boldsymbol{while}\ \ \theta > 0\ \boldsymbol{do}$
    $Apply\ r_a\ to\ find\ a\ subset\ \Psi \subseteq C\ of\ customers\ for\ removal$
    $\mathcal{L} \longleftarrow \mathcal{L} \cup \Psi$
    $Remove\ each\ customer\ c \in \Psi\ from\ S_C$
    $\boldsymbol{if}\ c \in C^P \boldsymbol{then}$
        $Remove\ external\ vendor\ \boldsymbol{v} \in V_c\ if\ service\ feasibility\ not\ violated$
    $\theta = \theta - 1$
$\boldsymbol{end\ while}$
Return $S_P\ and\ \mathcal{L}$

A simple removal procedure is illustrated in Figure 4.1. In this illustration, current solution is given in Figure 4.1a with an empty removal list. There are two regular customers where $C^R = \{C1, C2\}$, two premium customers where $C^P = \{C3, C4\}$, and two vendors where $V = \{V1, V2\}$. Premium customer $C3$ gets service from both vendors, i.e. $V_{C3} = \{V1, V2\}$, whereas premium customer $C4$ gets service only from vendor $V1$, i.e. $V_{C4} = \{V1\}$. Figure 4.1b illustrates the partial feasible solution after regular customer $C2$ is removed from the current feasible solution and put into the removal list. Figure 4.1c represents the partial feasible solution after premium customer $C3$ is removed together with vendor $V2$ from the solution given in Figure 4.1b. Notice that, since vendor $V1$ is responsible to serve both premium customers $C3$ and $C4$, it cannot be removed as a result of preserving service feasibility.

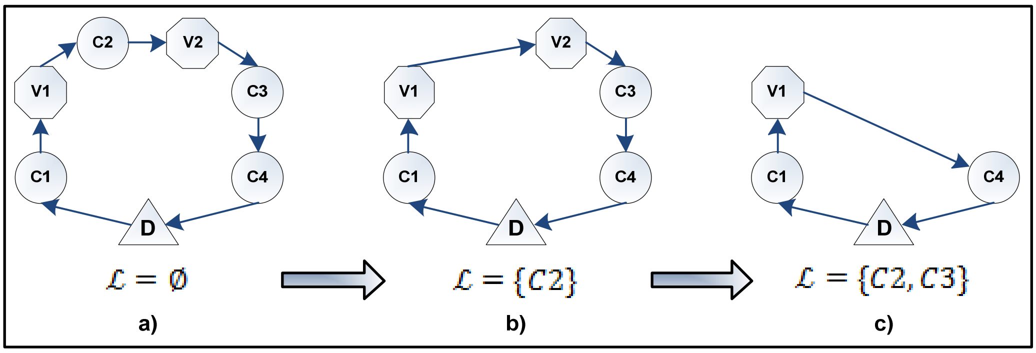


Figure 4.1: A simple illustration for removal process. **a)** Current solution **b)** Regular customer *C2* is removed. **c)** Premium customer *C3* is removed.

The removal algorithms used in the proposed ALNS framework are described below. The first ten are close adaptations of some removal operators presented in Ropke and Pisinger (2006a, 2006b), Pisinger and Ropke (2007), and Demir et al. (2012) whereas the next two involve improvements to the existing algorithms and the last is a new algorithm.

1. ***Random Removal (RR)***: This algorithm simply selects n_c customers at random to diversify the search mechanism and then removes them from S_C one by one.

2. ***Worst-Distance Removal (WDR)***: At the beginning of each removal iteration, remaining customers in S_C are ordered in non-increasing order of their *worst removal costs* defined with a function for customer $j \in C$ as $f_j = |c_{ij} + c_{jk}|$, i.e. the sum of distances from the preceding node $i \in N$ and following node $k \in N$ on the route. If O is the ordered array list of customers in this way, then the algorithm removes customer $j = O[\lfloor \lambda^{\kappa}|O| \rfloor]$ from S_C in the corresponding iteration, where $\lambda \in [0,1]$ is a random number and $\kappa \geq 1$ is a parameter called *worst removal determinism factor* that introduces some randomness in the selection of customers to avoid removing the same customers over and over again. A low value of κ yields too much randomness (Ropke and Pisinger, 2006a). O is updated after each node removal and the algorithm continues until n_c customers are removed from S_C.

3. ***Worst-Time Removal (WTR)***: In principle, this algorithm is similar to *WDR*; however, it considers deviation of service start time from beginning time window of each customer. Therefore, the *worst removal cost* function for customer $j \in C$

is now defined as $f_j = |\beta_j - beg_j|$, where β_j is the service start time at customer j. If O is the array list of customers in non-increasing order of these f_j values, then *WTR* algorithm also removes customer $j = O[\lfloor \lambda^{\kappa}|O| \rfloor]$ from S_C in the corresponding iteration and updates O after each node removal.

4. ***Shaw Removal (SR)***: This algorithm was introduced by Shaw (1998) in an attempt to identify more or less related customers, which are reasonably easy to change. The relevancy of customers $i, j \in C$ is defined by a *relatedness measure* $\Gamma(i,j)$ as follows,

$$\Gamma(i,j) = \phi_1 c_{ij} + \phi_2 |beg_i - beg_j| + \phi_3 \Omega_{ij} + \phi_4 |\delta_i - \delta_j| \tag{24}$$

where $\phi_1 - \phi_4$ are normalized weights named as *Shaw parameters* and

$$\Omega_{ij} = \begin{cases} -1, if customers i and j are in the same route \\ 1, otherwise \end{cases} \tag{25}$$

$$\delta_i = \begin{cases} d_i & \text{, if } i \in C^R \\ d_i + \sum_{p \in P} b_{ip} q_{ip} w_p & \text{, if } i \in C^P \end{cases} \tag{26}$$

The similarity of customers increases as $\Gamma(i,j)$ decreases. The algorithm initially starts with a randomly selected customer and adds it into removal list $\mathcal{L}$. In the subsequent removal iterations, it orders remaining customers in non-decreasing order of their *relatedness measures* with the already removed customer. If O is the ordered array list of remaining customers in this way, then the algorithm removes customer $j = O[\lfloor \lambda^{\eta}|O| \rfloor]$ from S_C in the corresponding iteration, where $\lambda \in [0,1]$ is a random number and $\eta \geq 1$ is a parameter called *Shaw removal determinism factor* like κ introduced for *WDR* and *WTR* algorithms.

5. ***Proximity-based Removal (PR)***: This algorithm is a special adaptation of *Shaw Removal* and it only considers distance-based relatedness of customers, i.e. $\phi_1 = 1$ and $\phi_2 = \phi_3 = \phi_4 = 0$ in (24).

6. ***Time-based Removal (TR)*****:** This algorithm is another special adaptation of *Shaw Removal* and it just considers time-based relatedness of customers, i.e. $\phi_2 = 1$ and $\phi_1 = \phi_3 = \phi_4 = 0$ in (24).

7. ***Demand-based Removal (DR)*****:** This algorithm is also a special adaptation of *Shaw Removal* and it considers demand-based relatedness of customers, i.e. $\phi_4 = 1$ and $\phi_1 = \phi_2 = \phi_3 = 0$ in (24).

8. ***Historical Knowledge Node Removal (HR)*****:** Unlike the other removal algorithms presented in this chapter, this heuristic makes use of historical information when removing customers. Let $f_j^l = c_{ij} + c_{jk}$ be *position cost* of customer $j \in C$ at iteration $l = 1 \dots N_I$, where $i \in N$ and $k \in N$ are preceding and succeeding nodes of customer j respectively. f_j^l value is calculated for each customer j at every iteration l and the best position cost f_j^* of customer j at iteration l is obtained by $f_j^* = min_{m=1,\dots,l-1}\{f_j^m\}$. If the *HR* algorithm is selected at iteration l, then it removes customer j^* which has the maximum deviation from its best position cost, i.e. $j^* = argmax_{j \in C}\{f_j^l - f_j^*\}$ and add it to removal list $\mathcal{L}$. After that the corresponding route of the already removed customer is updated and the next customer j^* is found over the recent S_P. The procedure continues until n_c customers are removed from S_C.

9. ***Neighborhood Removal (NR)*****:** Let $\mathbb{R}$ be the set of all routes in S_C and $\mathcal{R} = \{\sigma_1, \sigma_2, \dots, \sigma_{|\mathcal{R}|}\}$ be a route in $\mathbb{R}$. The *NR* algorithm calculates an average distance as $\bar{c}_{\mathcal{R}} = \left(\sum_{i=1}^{|\mathcal{R}|-1} c_{\sigma_i \sigma_{i+1}}\right)/|\mathcal{R}|$ for each $\mathcal{R} \in \mathbb{R}$ and removes a customer $j^* = argmax_{\mathcal{R} \in \mathbb{R}; j \in \mathcal{R}}\{\bar{c}_{\mathcal{R}} - c_{\mathcal{R} \setminus \{j\}}\}$. In other words, a customer which has the maximum deviation from the average distance of a route $\mathcal{R} \in \mathbb{R}$ is removed at each removal iteration and added to removal list $\mathcal{L}$. The procedure continues until n_c customers are removed from S_C.

10. ***Node Neighborhood Removal (NNR)*****:** This algorithm is a plain adaptation of the node neighborhood removal operator used in Demir et al. (2012). The *NNR* algorithm in this study simply selects a random customer $j \in C$ and then removes customer j along with the closest $n_c - 1$ customers around it.

11. ***Route Removal (RoR)***: The logic behind this algorithm is to completely remove at least one route from the current feasible solution. Let $\mathfrak{R} = \{\mathcal{R}_1, \dots, \mathcal{R}_i\} \subseteq \mathbb{R}$ be the set of randomly selected and removed routes from S_C and $n_c^{\mathfrak{R}} = n_c^{\mathcal{R}_1} + \cdots + n_c^{\mathcal{R}_i}$ be the total number of customers on these routes, where $\mathcal{R}_i$ represents the i^{th} randomly selected route for which $n_c^{\mathfrak{R}}$ becomes greater than or equal to n_c for the first time. Depending on $|\mathfrak{R}|$, *RoR* algorithm is divided into two subcategories: (i) *Single Route Removal* (*SRoR*) in which $\mathfrak{R} = \{\mathcal{R}_1\}$ and $n_c^{\mathfrak{R}} = n_c^{\mathcal{R}_1}$ and (ii) *Multiple Route Removal* (*MRoR*) in which $\mathfrak{R} = \{\mathcal{R}_1, \dots, \mathcal{R}_i\}$ and $n_c^{\mathfrak{R}} \geq n_c$. *MRoR* algorithm iteratively continues to remove all randomly selected routes in $\mathfrak{R}$. Unlike *MRoR*, *SRoR* algorithm does not guarantee removal of at least n_c customers. It is decided whether *SRoR* or *MRoR* algorithm should be used by means of a roulette wheel mechanism similar to (23) when *RoR* heuristic is adaptively selected. The *RoR* algorithm repeatedly selects a customer from the routes in $\mathfrak{R}$ until $n_c^{\mathfrak{R}}$ customers are completely removed from S_C.

12. ***Zone Removal (ZR)***: This algorithm improves the zone removal operator used in Demir et al. (2012). It simply removes a set of customers in a predefined area in the Cartesian coordinate system which is extracted from the given distribution network. The corner points of this region are computed by means of an initial preprocessing and the whole area is then divided into n_Z smaller pieces, named as *zones*, with respect to a predefined *zone direction*. Depending on this splitting direction, *ZR* algorithm is divided into four subcategories: (i) *Horizontal Zone Removal* (*HZR*), (ii) *Vertical Zone Removal* (*VZR*), (iii) *Right-Sided Zone Removal* (*RZR*), and (iv) *Left-Sided Zone Removal* (*LZR*). These subcategories are illustrated in Figure 4.2. Using a roulette wheel mechanism similar to (23) the *ZR* heuristic adaptively determines the sub-method to be used. Then, according to the selected sub-method the set of preprocessed zones, denoted by $\mathbb{Z}$, is considered and the customers are removed from randomly selected zones. Let $\mathcal{Z} = \{Z_1, \dots, Z_i\} \subseteq \mathbb{Z}$ be the set of these randomly selected zones and $n_c^{\mathcal{Z}} = n_c^{Z_1} + \cdots + n_c^{Z_i}$ be the total number of customers on these zones, where Z_i represents the i^{th} randomly selected zone for which $n_c^{\mathcal{Z}}$ becomes greater than or equal to n_c for the first time. The *ZR* algorithm iteratively removes all customers in each zone $Z \in \mathcal{Z} \backslash \{Z_i\}$. On the other hand, customers in zone Z_i are sorted in non-decreasing

order of their distance to the centroid of zone Z_i and the closest $\left(n_c - \sum_{k=1}^{i-1} n_c^{Z_k}\right)$ customers are removed from zone Z_i.

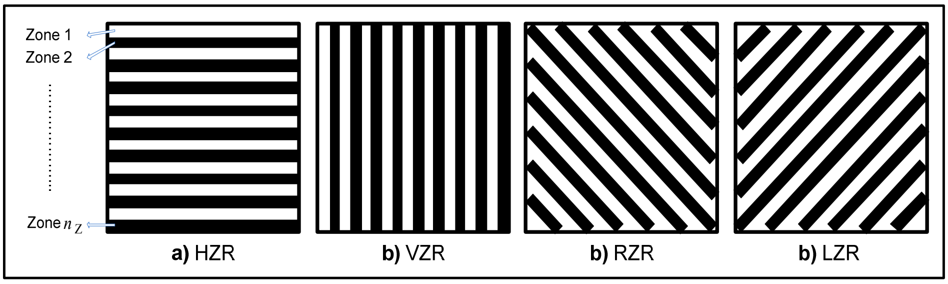


Figure 4.2: Zone Removal sub-methods

13. ***Route Neighborhood Removal (RNR)*:** The main idea of this algorithm is to integrate some feasibility concerns into a distance-based procedure to increase the efficiency of the removal process in the following insertion algorithms. Let $\mathbb{R} = \{\mathcal{R}_1, \dots, \mathcal{R}_{|\mathbb{R}|}\}$ be an unordered set of all routes in the current feasible solution, $\mathcal{R}_i = \{i_1, \dots, i_{|\mathcal{R}_i|}\} \in \mathbb{R}$ and $\mathcal{R}_j = \{j_1, \dots, j_{|\mathcal{R}_j|}\} \in \mathbb{R}$ where $\mathcal{R}_i \neq \mathcal{R}_j$. Assume that we select a customer $i_m \in \mathcal{R}_i$ and a node $j_n \in \mathcal{R}_j$ to create an *eligible node pair* (i_m, j_n) in such a way that the modified route $\mathcal{R}_i^* = \{i_1, \dots, i_m, j_n, i_{m+1}, \dots, i_{|\mathcal{R}_i^*|}\}$ still ensures time window as well as load feasibility and $\left(c_{i_m, j_n} + c_{j_n, i_{m+1}}\right)$ is minimized. At each removal iteration, the *RNR* algorithm randomly selects a route $\mathcal{R}_i \in \mathbb{R}$ and then searches for an *eligible node pair* (i_m, j_n). If the algorithm is able to find such a pair, then j_n is removed from $\mathcal{R}_j$ and added to the removal list $\mathcal{L}$. Otherwise, a customer $i_r \in \mathcal{R}_i$ is randomly selected and removed from $\mathcal{R}_i$. This algorithm lasts $\theta = n_c$ iterations with the updated routes at the end of each iteration and node-to-node distance matrix is generated with an initial preprocessing to increase efficiency. The working mechanism of this algorithm is illustrated in Figure 4.3. There are three routes given by $\mathbb{R} = \{R1, R2, R3\}$. In Figure 4.3a, route $R2$ is randomly selected as $\mathcal{R}_i$. Then, an eligible node pair (i_m, j_n), where $i_m \in R2$ and $j_n \in (R1 \text{ or } R3)$, is searched. Such a pair is found for $i_1 = 1$ and $j_1 = 1$ when $\mathcal{R}_j = R1$. Then $j_1 = 1$ is removed from route $R1$ and the solution is updated as in Figure 4.3b.

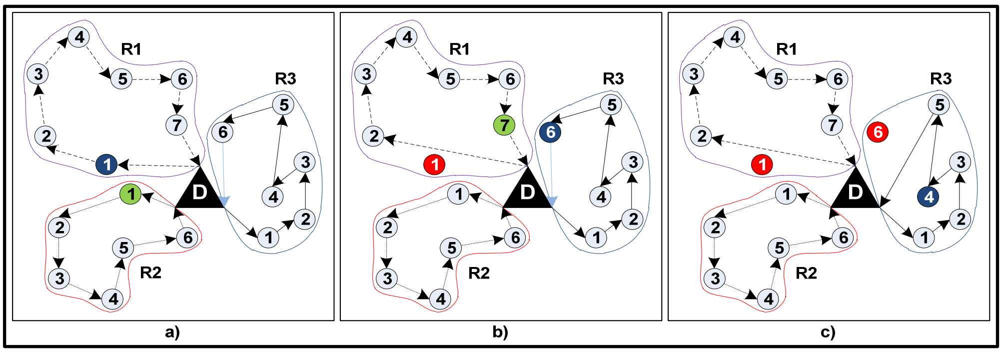


Figure 4.3: Route Neighborhood Removal Algorithm. **a)** R_i= $R2$ **b)** R_i= $R1$ **c)** R_i= $R3$

In Figure 4.3b, this time route $R1$ is randomly selected as $\mathcal{R}_i$ and the next eligible node pair is searched. Such a pair is found for $i_6 = 7$ and $j_6 = 6$ when $\mathcal{R}_j = R3$. Then $j_6 = 6$ is removed from route $R3$ and the solution is updated as in Figure 4.3c. In Figure 4.3c, this time route $R3$ is randomly selected as $\mathcal{R}_i$ and the next eligible node pair is searched, however, such a pair does not exist. So $i_4 = 4$ is randomly selected to be removed.

4.1.2 Vendor Selection/Allocation Algorithms

The algorithms presented in this section are indispensable auxiliary actors in the repair phase of the proposed ALNS framework. Let $V_j \subseteq V$ be a set of external vendors responsible to supply premium product orders set $P_j = \left\{p_1, \dots, p_{|P_j|}\right\} \subseteq P$ demanded by premium customer $j \in C^P$ and $P_v \subseteq P$ be a set of premium products provided by vendor $v \in V$. Also, let O_i^p be a sorted list of all eligible vendors $v \in V$, i.e. vendors for which premium product $p \in P_v$, in non-decreasing order of their distance to node $i \in N$. These O_i^p lists are generated with an initial preprocessing to increase efficiency of the distance-based vendor selection/allocation algorithms. Since V_j of each removed premium customer $j \in (C^P \cap \mathcal{L})$ is destroyed during the removal process, the vendor selection algorithms are used to allocate a new proper set V_j to each one of them at the beginning of its insertion process. We now describe six new vendor selection/allocation algorithms used in our study.

1. ***Node Neighborhood Vendor Selection (NNVS)*:** The main purpose of this algorithm is to ensure that each removed premium customer gets its service from

the closest eligible vendors in the temporary feasible solution S_T. In other words, the algorithm generates a new feasible set V_j for each customer $j \in (C^P \cap \mathcal{L})$ in such a way that the maximum distance to customer j from a vendor $v \in V_j$, i.e. $d(V_j) = max_{v \in V_j}\{c_{v,j}\}$, is minimized. A pseudo-code of the *NNVS* heuristic is presented in Algorithm 4.2. The algorithm starts with a set of input parameters mostly related to target customer and Steps 3-15 are repeated for each $p \in P_j$ to get a new feasible V_j that has the least $d(V_j)$ amount for the target customer.

Algorithm 4.2: The generic structure of the *NNVS* algorithm

	Input: $j \in (C^P \cap \mathcal{L}), P_j, P_v\ for\ each\ v \in V, and\ O_j^p\ for\ each\ p \in P_j$
1	$Initialize\ V_j \longleftarrow \emptyset$ and $d(V_j) = \infty$
2	$\boldsymbol{for}\ each\ p \in P_j \boldsymbol{do}$
3	$\quad Initialize\ \mathcal{H}, set\ of\ premium\ products\ not\ yet\ supplied, \mathcal{H} \longleftarrow P_j$
4	$\quad Initialize\ temporary\ output\ \Upsilon_j\ for\ V_j, \Upsilon_j \longleftarrow \emptyset$
5	$\quad Select\ the\ closest\ vendor\ \boldsymbol{v}\ supplying\ \boldsymbol{p}, i.e. vendor\ v = O_j^p[0]$
6	$\quad \Upsilon_j \longleftarrow \Upsilon_j \cup \{v\}$
7	$\quad \mathcal{H} \longleftarrow \mathcal{H} \backslash (\mathcal{H} \cap P_v)$
8	$\quad \boldsymbol{while}\ \mathcal{H} \neq \emptyset\ \boldsymbol{do}$
9	$\qquad Select\ an\ unsupplied\ premium\ product\ p_u \in \mathcal{H}$
10	$\qquad Select\ vendor\ v = O_j^{p_u}[0]$
11	$\qquad \Upsilon_j \longleftarrow \Upsilon_j \cup \{v\}$
12	$\qquad \mathcal{H} \longleftarrow \mathcal{H} \backslash (\mathcal{H} \cap P_v)$
13	$\quad \boldsymbol{end\ while}$
14	$\quad \boldsymbol{if}\ d(\Upsilon_j) < d(V_j) \boldsymbol{then}$
15	$\qquad V_j \longleftarrow \Upsilon_j$
16	$\boldsymbol{end\ for}$
17	**Return** V_j

2. ***Route Neighborhood Vendor Selection (RNVS)*****:** The logic behind this algorithm is to consider already existing routes in the partial feasible solution S_P while generating a new feasible V_j for each removed premium customer $j \in (C^P \cap \mathcal{L})$. Let $\mathbb{R} = \{\mathcal{R}_1, \dots, \mathcal{R}_{|\mathbb{R}|}\}$ be an unordered set of all routes in S_P and $\mathcal{R}_i = \{i_1, \dots, i_{|\mathcal{R}_i|}\}$ be a route with unsorted nodes in $\mathbb{R}$. The *RNVS* algorithm has a potential to produce different results for each route $\mathcal{R}_i \in \mathbb{R}$ since it aims to identify new external vendor sets whose elements are located around $\mathcal{R}_i$. In other

words, it forms a new set V_j around a route $\mathcal{R}_i$ for the removed premium customer j such that $d(V_j) = max_{v \in V_j}\left\{min_{i \in \mathcal{R}_i}\{c_{v,i}\}\right\}$ is minimized. Let $v_i = O_i^p[0]$ be the closest vendor to node $i \in \mathcal{R}_i$ which supplies premium product $p \in P_j$. The *RNVS* heuristic takes $\mathcal{R}_i$ and O_i^p for each $p \in P_j$ and $i \in \mathcal{R}_i$ as additional inputs and only differs in Step 5 and Step 10 of the pseudo-code given in Algorithm 4.2. In those steps, the closest vendor v is selected as $v = argmin_{i \in \mathcal{R}_i;\, v_i \in V_j}\{c_{v_i,i}\}$ for the *RNVS* algorithm. A simple example related to the desired outputs of *NNVS* and *RNVS* algorithms are given in Figure 4.4 where a current feasible solution S_C is presented in Figure 4.4a. In this setting, there are seven regular customers where $C^R = \{C1, C2,\ C3, C4, C5, C7, C8\}$, one premium customer where $C^P = \{C6\}$, five external vendors where $V = \{V1, V2, V3, V4, V5\}$, and two premium products where $P = \{1,2\}$. Vendor $V3$ supplies both premium products, i.e. $P_{V3} = \{1,2\}$, vendors $V1$ and $V4$ supply only the first premium product, i.e. $P_{V1} = P_{V4} = \{1\}$, and vendors $V2$ and $V5$ supply only the second premium product, i.e. $P_{V2} = P_{V5} = \{2\}$. Premium customer $C6$ orders both premium products, i.e. $P_{C6} = \{1,2\}$, and gets service from vendor $V3$ for his entire order, i.e. $V_{C6} = \{V3\}$.

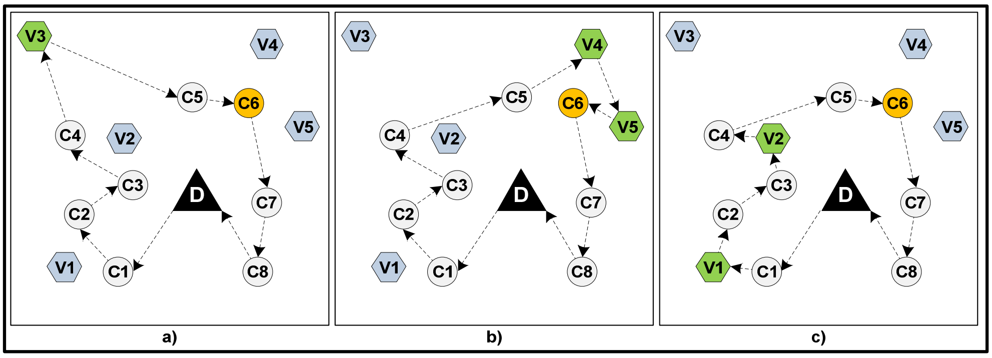


Figure 4.4: A simple example related to the desired outputs of NNVS and RNVS algorithms. **a)** Current feasible solution S_C. **b)** NNVS algorithm is selected in the repair phase of S_P. **c)** RNVS algorithm is selected in the repair phase of S_P.

Assume that the premium customer $C6$ is removed together with $V3$ from S_C given in Figure 4.4a and the partial feasible solution S_P is obtained. If *NNVS* algorithm is selected in the repair phase of S_P, then $V_{C6} = \{V4, V5\}$ will be assigned to premium customer $C6$. A possible insertion of $\{V4, V5, C6\}$ into S_P is given in Figure 4.4b. On the other hand, if *RNVS* algorithm is selected in the repair phase of S_P, then

$\mathrm{V_{C6}} = \{\mathrm{V1}, \mathrm{V2}\}$ will be assigned to premium customer $\mathrm{C6}$. A possible insertion of $\{\mathrm{V1}, \mathrm{V2}, \mathrm{C6}\}$ into $\mathrm{S_P}$ is given in Figure 4.4c.

3. ***Node Neighborhood Vendor Selection with Noise (NNVSN)*:** This algorithm is a special adaptation of the *NNVS* heuristic. It takes into account a degree of freedom in the closest vendor allocation processes to avoid selection of the same vendor sets over and over again. Therefore, Step 5 and Step 11 of Algorithm 4.2 is modified to select the closest vendor $v = O_j^p\left[\left\lfloor \lambda \left|O_j^p\right| \psi \right\rfloor\right]$, where $\lambda \in [0,1]$ is a random number and $\psi \in [0,1]$ is a parameter called *freedom percentage* for vendor selection/allocation algorithms.

4. ***Route Neighborhood Vendor Selection with Noise (RNVSN)*:** The aim of this algorithm is to integrate a degree of freedom into the closest vendor selection processes of the *RNVS* heuristic. By considering other parameter definitions as they are in the *RNVS* and *NNVSN* algorithms, let $v_i = O_i^p\left[\left\lfloor \lambda \left|O_i^p\right| \psi \right\rfloor\right]$ be the $\left(\left\lfloor \lambda \left|O_i^p\right| \psi \right\rfloor + 1\right)^{th}$ closest vendor to node $i \in \mathcal{R}_i$ which supplies premium product $p \in P_j$. Then, the closest vendor v is selected as $v = argmin_{i \in \mathcal{R}_i; v_i \in V_j}\{c_{v_i,i}\}$ during the *RNVSN* algorithm.

5. ***Random Vendor Selection (RVS)*:** This algorithm randomly allocates a new external vendor set to each removed premium customer to diversify the vendor selection procedure. The pseudo-code is given in Algorithm 4.3.

Algorithm 4.3: The generic structure of the *RVS* algorithm

Input: $j \in (C^P \cap \mathcal{L}), P_j, and\ P_v\ for\ each\ v \in V$

$Initialize\ V_j, V_j \longleftarrow \emptyset$
$Initialize\ \mathcal{H}, set\ of\ premium\ products\ not\ yet\ supplied, \mathcal{H} \longleftarrow P_j$
$\boldsymbol{while}\ \mathcal{H} \neq \emptyset\ \boldsymbol{do}$
    $Select\ a\ random\ vendor\ \boldsymbol{v} \in V\ for\ which\ (\mathcal{H} \cap P_v) \neq \emptyset$
    $V_j \longleftarrow V_j \cup \{v\}$
    $\mathcal{H} \longleftarrow \mathcal{H} \backslash (\mathcal{H} \cap P_v)$
$\boldsymbol{end\ while}$

Return V_j

6. ***Historical Knowledge Vendor Selection (HVS)*****:** This algorithm makes use of historical information when generating new feasible external vendor sets. Let $V_j^l \subseteq V$ be a set of external vendors in S_T responsible to supply premium product orders demanded by premium customer $j \in C^P$ at iteration $l = 1 \dots N_I$. Also let $\mho_l = V_j^l \cup \{j\}$ and $f_j^l = \sum_{z \in \mho_l} \left(c_{z_p,z} + c_{z,z_s} \right)$ be *service cost* of premium customer j at iteration l, where $z_p \in N$ and $z_s \in N$ are predecessor and successor of node $z \in \mho_l$ respectively. Service costs are calculated for each premium customer in every iteration. If the *HVS* algorithm is selected in iteration$l$, then the external vendor set V_j with the best known service cost until iteration l, i.e. $V_j = argmin_{m=1,\dots,l-1}\{f_j^m\}$, is allocated to each removed premium customer $j \in (C^P \cap \mathcal{L})$.

4.1.3 Insertion Algorithms

In the repair phase of the proposed ALNS framework, the insertion algorithms presented in this section are responsible for deriving a temporary feasible solution S_T by placing each customer in the removal list back into the partially destroyed feasible solution S_P. Throughout this process, time windows and capacity feasibilities are considered for regular customers whereas additional service feasibility, i.e. external vendor set of each premium customer being visited before the customer in the corresponding route, is taken into account for premium customers. Each removed customer is placed into a new route if one of these feasibility concerns is not satisfied by means of the existing routes in S_P. The insertion algorithms described below are all sequential. The first three are partially adapted from Ropke and Pisinger (2006a), Pisinger and Ropke (2007) or Demir et al. (2012), whereas the last one introduces a new and different perspective.

1. ***Greedy Insertion (GI)*****:** This algorithm is a simple construction heuristic that inserts each removed customer into the best possible position of a route in S_P. Let $\mathbb{R} = \{\mathcal{R}_1, \dots, \mathcal{R}_{|\mathbb{R}|}\}$ be an unordered set of all routes in S_P and $\mathcal{R}_i = \{i_1, \dots, i_{|\mathcal{R}_i|}\}$ be a route in $\mathbb{R}$ where i_k is the node in the k^{th} position of the route. Also, let $\Pi_j^{\mathcal{R}_i} = min_{i_k \in \mathcal{R}_i}\{c_{i_k,j} + c_{j,i_{k+1}} - c_{i_k,i_{k+1}}\}$ be the *additional cost* of inserting a node $j \in N \backslash \{C^P\}$ into route $\mathcal{R}_i$ at the position that yields the least increase in the

objective value. Each regular customer $c_r \in (C^R \cap \mathcal{L})$ is inserted into route $\mathcal{R}_i$ by just itself whereas each premium customer $c_p \in (C^P \cap \mathcal{L})$ is inserted together with its new external vendor set V_{c_p} in such a way that each vendor $v \in V_{c_p}$ takes place in route $\mathcal{R}_i$ before premium customer c_p to assure service feasibility. For ease of implementation we use an identical copy Λ_i of route $\mathcal{R}_i$ to determine the best insertion positions of a premium customer c_p and its corresponding vendors. At first, a vendor $v \in V_{c_p}$ is randomly selected and it is inserted into its best position in Λ_i with an additional cost of $\Pi_v^{\Lambda_i}$. This procedure continues until each vendor $v \in V_{c_p}$ is inserted into route Λ_i which is simultaneously updated with each insertion. Then, we acquire the position $\varphi_v \in \{1, \dots, |\Lambda_i|\}$ of each vendor $v \in V_{c_p}$ in the route Λ_i. Finally, premium customer c_p is inserted into route Λ_i with an individual additional cost of $\mathbb{P}_{c_p}^{\Lambda_i} = min_{i_k \in \Lambda_i\, ;\, k \geq max_{v \in V_{c_p}}\{\varphi_v\}} \left\{ c_{i_k, c_p} + c_{c_p, i_{k+1}} - c_{i_k, i_{k+1}} \right\}$. Consequently, the total additional cost of inserting premium customer c_p into route $\mathcal{R}_i$ is given by $\Pi_{c_p}^{\mathcal{R}_i} = \mathbb{C}_{c_p}^{\Lambda_i} + \mathbb{P}_{c_p}^{\Lambda_i}$, where $\mathbb{C}_{c_p}^{\Lambda_i} = \sum_{v \in V_{c_p}} \Pi_v^{\Lambda_i}$ is the *vendor coverage cost* of premium customer c_p in route Λ_i. If node $j \in N$ cannot be inserted into route $\mathcal{R}_i$, then $\Pi_j^{\mathcal{R}_i}$ is set to infinity. We then define $\Pi_j = min\left\{\Pi_j^{\mathcal{R}_0}, min_{\mathcal{R}_i \in \mathbb{R}}\left\{\Pi_j^{\mathcal{R}_i}\right\}\right\}$ as the *insertion cost* of putting node j into its minimum cost position overall, where $\Pi_j^{\mathcal{R}_0}$ is the cost of inserting node j into a new route by just itself. The *GI* algorithm selects a customer $j^* = argmin_{j \in \mathcal{L}}\{\Pi_j\}$ and removes it from $\mathcal{L}$ by inserting customer j^* and each vendor $v \in V_{j^*}$, if any, at their minimum cost positions. This procedure is repeated until removal list $\mathcal{L}$ becomes empty. The pseudo-code of the *GI* algorithm is presented in Algorithm 4.4. Both *Clean* and *Noise Imposed* versions of *GI* algorithm are used by considering $\Pi_j^{\mathcal{R}_i}$ values as the $actual\ cost$.

2. ***Regret-k Insertion (R-kI)*:** Since the *GI* algorithm generally delays the insertion of the customers with large *insertion costs* this algorithm tries to overcome this myopic behavior by considering a sort of look ahead information to decide which customer to insert next (Ropke and Pisinger, 2006a). Let $r_{jk} \in \{1, \dots, |\mathbb{R}|\}$ correspond to the route for which customer $j \in \mathcal{L}$ has the k^{th} lowest insertion

cost, i.e. $\Pi_j^{r_{jk}} \leq \Pi_j^{r_{jk'}}$ for $k \leq k'$. We then, define a *regret-k value* Ч_j^k for each customer $j \in \mathcal{L}$ as the difference in the cost of inserting customer j in its best route and its k^{th} best route, i.e. $\text{Ч}_j^k = \sum_{i=1}^{k}\left(\Pi_j^{r_{ji}} - \Pi_j^{r_{j1}}\right)$. The *R-kI* algorithm selects a customer $j^* = argmax_{j\in\mathcal{L}}\{\text{Ч}_j^k\}$ to insert customer j^* and each vendor $v \in V_{j^*}$, if any, at their minimum cost positions. In case of a tie, the customer with lowest insertion cost is selected. In this study, we consider both *Clean* and *Noise Imposed* versions of *Regret-2*, *Regret-3*, *Regret-4*, and *Regret-m* insertion algorithms, where $m = |\mathbb{R}|$.

Algorithm 4.4: The generic structure of the *GI* algorithm

Input: $i_a \in A_I, v_a \in A_V, S_P, \mathcal{L}$

$\boldsymbol{while}\ \mathcal{L} \neq \emptyset\ \boldsymbol{do}$
    $\boldsymbol{for}\ each\ regular\ customer\ c_r \in (C^R \cap \mathcal{L})$
        $Calculate\ \Pi_{c_r}^{\mathcal{R}_i}\ for\ each\ route\ \mathcal{R}_i \in \mathbb{R} \cup \{\mathcal{R}_0\}$
        $Get\ \Pi_{c_r} = min_{\mathcal{R}_i \in \mathbb{R}\cup\{\mathcal{R}_0\}}\{\Pi_{c_r}^{\mathcal{R}_i}\}$
    $\boldsymbol{end\ for}$
    $\boldsymbol{for}\ each\ premium\ customer\ c_p \in (C^P \cap \mathcal{L})$
        $\boldsymbol{for}\ each\ route\ \mathcal{R}_i \in \mathbb{R} \cup \{\mathcal{R}_0\}$
            $Apply\ v_a\ to\ generate\ a\ new\ feasible\ V_{c_p}\ wrt.\ \mathcal{R}_i$
            $Initialize\ \Lambda_i, identical\ copy\ of\ route\ \mathcal{R}_i, \Lambda_i \longleftarrow \mathcal{R}_i$
            $Initialize\ \mathcal{H}, set\ of\ vendors\ not\ yet\ inserted, \mathcal{H} \longleftarrow V_{c_p}$
            $\boldsymbol{while}\ \mathcal{H} \neq \emptyset\ \boldsymbol{do}$
                $Select\ a\ random\ vendor\ \boldsymbol{v} \in \mathcal{H}\backslash\Lambda_i\ and\ Calculate\ \Pi_v^{\Lambda_i}$
                $Insert\ vendor\ \boldsymbol{v}\ into\ its\ best\ position\ in\ \Lambda_i$
                $\mathcal{H} \longleftarrow \mathcal{H}\backslash\{v\}$
            $\boldsymbol{end\ while}$
            $Calculate\ \mathbb{C}_{c_p}^{\Lambda_i}\ and\ \mathbb{P}_{c_p}^{\Lambda_i}$
        $\boldsymbol{endfor}$
        $Get\ \Pi_{c_p} = min_{\mathcal{R}_i \in \mathbb{R}\cup\{\mathcal{R}_0\}}\left\{\Pi_{c_p}^{\mathcal{R}_i}\right\}$
    $\boldsymbol{endfor}$
    $Select\ customer\ c^* = argmin_{c\in\mathcal{L}}\{\Pi_c\}$
    $Insert\ customer\ c^* and\ V_{c^*}, if\ any, into\ their\ best\ positions\ in\ S_P$
    $\mathcal{L} \longleftarrow \mathcal{L}\backslash\{c^*\}$
$\boldsymbol{end\ while}$

return S_T

3. ***Zone Insertion (ZI)*****:** This algorithm is a modified version of the zone insertion operator used in Demir et al. (2012). When determining the best insertion position of each node, it simply gives priority to time window instead of the distance with an aim to leave enough space for future insertions. Let $\mathbb{W}_j^{\mathcal{R}_i} = min_{i_k \in \mathcal{R}_i}\left\{max\{0, \left(T_{i_k} + s_{i_k} + t_{i_k j}\right) - beg_j\}\right\}$ be the *waiting cost* of inserting a node $j \in N\backslash\{C^P\}$ into route $\mathcal{R}_i$ at a feasible position that yields the least waiting time. Total waiting cost of a premium customer $j \in (C^P \cap \mathcal{L})$ is calculated in a similar way given in the *GI* heuristic. The algorithm adaptively selects one of the zone directions mentioned in the *ZR* heuristic and works with the corresponding set of zones $\mathbb{Z} = \left\{Z_1, \dots, Z_{n_Z}\right\}$. At each iteration, the *ZI* algorithm randomly selects a customer $j \in \mathcal{L}$ located in zone Z_l. Then, it identifies the routes $\mathcal{R}_i^* \in \mathbb{R}$ for which $(Z_l \cap \mathcal{R}_i^*) \neq \emptyset$ and calculates *zone insertion cost* $\mathbb{W}_j = min_{\mathcal{R}_i^*}\left\{\mathbb{W}_j^{\mathcal{R}_i^*}\right\}$ as the cost of inserting customer j in its least waiting cost position. If no feasible position exists, i.e. $\mathbb{W}_j = \infty$, then customer j is inserted at its minimum cost position by means of the *GI* heuristic. Both *Clean* and *Noised Imposed* versions of the *ZI* algorithm are implemented.

4. ***Greedy Insertion with New Route Openings (GINO)*****:** This algorithm basically makes use of the *GI* heuristic with a slight difference in customer insertion policy. Let $\mathbb{R}_{S_P}$ and $\mathbb{R}_{S_B}$ be the set of all routes in S_P and S_B, respectively. Also, let ξ be an integer parameter that represents the *maximum route allowance*. At each iteration, the *GINO* algorithm selects a customer $j^* = argmin_{j \in \mathcal{L}}\{\Pi_j\}$; however, if $\left|\mathbb{R}_{S_P}\right| \leq \left|\mathbb{R}_{S_B}\right| + \xi$, then with a *new route opening probability* γ it inserts customer j^* and each vendor $v \in V_{j^*}$, if any, in a new route instead of their minimum cost positions.

The proposed ALNS approach with a simulated annealing (SA) local search framework at the master level is presented in Algorithm 4.5.

Algorithm 4.5: The generic structure of the *ALNS* algorithm with *SA*

Input: $A_R, A_V, A_I, N_I, N_{SI}, N_{IwI}, \mu, \varepsilon$

$Generate\ an\ S_I\ by\ using\ Greedy\ or\ Regret-2\ Heuristic$
$Initialize\ P^s_{r_a}, P^s_{v_a}, P^s_{i_a}\ for\ each\ r_a \in A_R, v_a \in A_V, and\ i_a \in A_I$
$Initialize\ P^s_a\ for\ each\ subalgorithm\ a \in r_a, and\ a \in i_a$
$Initialize\ T\ and\ T_{Start}\ by\ using\ z(S_I)\ and\ \mu$
$Let\ j\ be\ the\ outmost\ iteration\ counter\ initialized\ as\ j \longleftarrow 1$
$Let\ S_C \longleftarrow S_B \longleftarrow S_I$
$Let\ k\ be\ the\ counter\ of\ iterations\ without\ improvement\ in\ S_B, k \longleftarrow 1$
$\boldsymbol{while}\ j \leq N_I\ \boldsymbol{do}$
  $\boldsymbol{if}\ k \geq N_{IwI}\ \boldsymbol{then}$
    $Generate\ a\ new\ S_I\ by\ adaptively\ selecting\ Greedy\ or\ Regret\ Heuristic$
    $Let\ S_C \longleftarrow S_I$
  $Select\ a\ removal\ algorithm\ r_a \in A_R\ with\ probability\ P^s_{r_a}$
  $Generate\ S_P\ by\ applying\ algorithm\ r_a\ to\ S_C$
  $Select\ a\ vendor\ selection/allocation\ algorithm\ v_a \in A_V\ with\ probability\ P^s_{v_a}$
  $Select\ an\ insertion\ algorithm\ i_a \in A_I\ with\ probability\ P^s_{i_a}$
  $Generate\ S_T\ by\ applying\ i_a\ together\ with\ v_a\ to\ S_P$
  $\boldsymbol{if}\ z(S_T) < z(S_C)\ \boldsymbol{then}$
    $S_C \longleftarrow S_T$
  $\boldsymbol{else}$
    $Let\ v = exp\ (-(z(S_T) - z(S_C))/T)$
    $Generate\ a\ random\ number\ э \in [0,1]$
    $\boldsymbol{if}\ э < v\ \boldsymbol{then}$
      $S_C \longleftarrow S_T$
  $\boldsymbol{if}\ z(S_C) < z(S_B)\ \boldsymbol{then}$
    $S_B \longleftarrow S_T$
  $\boldsymbol{if}\ j \equiv 0\ \ (mod\ N_{SI})\ \boldsymbol{then}$
    $Update\ P^s_a\ probabilities\ using\ the\ adaptive\ weight\ procedure$
  $T \longleftarrow T\varepsilon$
  $j \longleftarrow j+1$
$\boldsymbol{end\ while}$

return S_B

Chapter 5

Computational Experiments

In this chapter, we perform extensive computational experiments to test the performance of the proposed ALNS approach. In order to validate the performance of the proposed algorithm, we first solve the well-known 100-node Solomon benchmark instances for VRPTW and compare the results against those published in the literature. Then, we generate benchmark instances by adapting 25-node, 50-node, and 100-node Solomon problems to VRPVSIPD setting and solve those using ALNS. For the 25-node instances, we compare our results with those obtained by using CPLEX. For larger instances, we present our results as benchmarks for future studies. We have performed all of our experiments on a computer equipped with Intel Core2 Quad 2.40 GHz CPU (Q6600) and 4 GB RAM. We have coded all the algorithms using the Java programming language with single precision floating point numbers for distances and travel times unless otherwise stated.

5.1 Parameter Tuning

Since there are not enough benchmark instances to evaluate the performance of our proposed heuristic, we first need to test it on certain underlying problems in order to increase and fine-tune its reliability. We consider nine VRPTW benchmark instances with 100 customers to tune the parameter values through extensive computational experiments. The instances selected are as follows: R104, R112, R201, R204, R207, RC104, RC106, RC206, and RC207. Although these instances do not involve vendors and premium customers, we think that that the values determined will likely perform well for the newly generated VRPVSIPD instances described in Section 5.3.1 as all parameters except the *freedom percentage* parameter ψ for vendor selection/allocation algorithms are common for both regular and premium customers. This parameter is adjusted by means of the modified versions of the tuning instances after all the other parameters have been set.

To tune the parameters we adopt a strategy similar to that of Ropke and Pisinger (2006a). First of all, for the parameters used in common we initially consider the values used in Ropke and Pisinger (2006a, 2006b), Pisinger and Ropke (2007), and Demir et al. (2012), whereas we determine a reasonable value for the remaining parameters. Next, we allow one parameter to take a number of predefined values by keeping the other parameters fixed. We then run the proposed ALNS heuristic ten times on the tuning instances using each parameter setting and we select the setting with the least average deviation from the best-known solutions. Once a parameter has been tuned, we consider the next one in a similar fashion until all parameters have been tuned. Since tuning a large number of parameters may require considerable amount of time, we stop the procedure after one pass; however, the final parameter setting may be further improved by repeating the whole process with the new set of parameters, though at the expense of additional computational effort. The tuned parameters and their final values are summarized in Table 5.1. The initial setting, tuning sequence, range and deviation of each parameter are given in Appendix A. The parameter tuning results show that most of the parameters have preserved their initial values.

Table 5.1: Parameters used in the proposed ALNS Heuristic

Parameter Description	**Parameter Value**
Maximum number of iterations (N_I)	25000
Number of iterations for roulette wheel (N_{SI})	100
MNOIWITRIFS [(*)] (N_{IwI})	4000
Roulette wheel reaction factor (ρ)	0.1
New global solution score (σ_1)	20
Better solution score (σ_2)	16
Worse solution score (σ_3)	13
Start temperature control parameter (μ)	0.05
Cooling rate (ε)	0.9998
Noise parameter ($\ni$)	0.025
Lower limit of the number of customers to remove ($\underline{n}_c$)	$min\{0.1\|N\|, 30\}$
Upper limit of the number of customers to remove ($\overline{n}_c$)	$min\{0.4\|N\|, 60\}$
First Shaw parameter (ϕ_1)	9
Second Shaw parameter (ϕ_2)	3
Third Shaw parameter (ϕ_3)	5
Fourth Shaw parameter (ϕ_4)	2
Determinism parameter for Shaw removal operators (η)	6
Determinism parameter for worst removal operators (κ)	3
Number of zones (n_Z)	11
New route opening probability (γ)	0.2
New route opening allowance (ξ)	2
Freedom percentage for vendor selection processes (ψ)	0.35

(*):Maximum number of iterations without improvement to refresh initial feasible solution

Similar to Ropke and Pisinger (2006a) and Demir et al. (2012) our sensitivity analysis indicates that 25000 iterations are enough to get good quality solutions in reasonable time periods. In each of these iterations, the number of customers to remove (n_c) is chosen randomly between a lower limit $\underline{n}_c = min\{0.1|N|, 30\}$ and an upper limit $\overline{n}_c = min\{0.4|N|, 60\}$. Experiments show that the performance of the insertion heuristics decreases as the upper limit $\bar{n}_c$ increases, whereas decreasing $\bar{n}_c$ mostly results in minor improvements. Since we introduce additional diversification mechanisms, our setting of the parameters σ_1, σ_2 and σ_3 is consistent with the expected setting $\sigma_1 \geq \sigma_2 \geq \sigma_3$ to reward an operator for good performance, unlike Ropke and Pisinger (2006a) and Demir et al. (2012) in which the discovery of a worse solution is rewarded more than the discovery of a better solution.

5.2 Experiments on VRPTW Instances

In this section, we solve the well-known 100-node Solomon benchmark instances for VRPTW and compare the results against those published in the literature.

5.2.1 Results on the Truncated Numbered Data

We first provide results on Solomon's VRPTW benchmark instances with 100 customers to evaluate the performance of our ALNS heuristic in comparison to the ALNS heuristics proposed by Pisinger and Ropke (2007) and Demir et al. (2012), which we denote by PR and DBL, respectively. Table 5.2 contains summary result of this comparison, whereas the detailed results are presented in Table B.1 of Appendix B. The comparisons are made in terms of the best solution and mean values obtained through 10 runs of each algorithm, where each run lasts for 25000 iterations. Distances and travel times are truncated (rounded down) to one decimal and travel distance is minimized. The optimal distances were obtained from Roberti (2012).

Table 5.2: Summary results of Solomon's truncated numbered VRPTW instances

	Optimal	PR			DBL		The Proposed ALNS			Deviation from (%)			
Class	distance	best dist.	avg. dist.	avg. time (secs)	best dist.	avg. time (secs)	best dist.	avg. dist.	avg. time (secs)	optimal	PR best	DBL best	PR avg.
R1	1173.6	1174.8	1177.4	32.8	1174.5	45.1	1174.1	1177.1	51.1	0.05	-0.06	-0.04	-0.03
C1	826.7	826.7	826.7	32.1	826.7	40.6	826.7	826.7	43.4	0.00	0.00	0.00	0.00
RC1	1334.5	1336.8	1342.2	30.3	1336.7	42.3	1334.7	1340.5	51.4	0.01	-0.16	-0.15	-0.14
R2	872.5	875.6	878.9	62.6	874.6	75.6	875.1	878.8	43.5	0.29	-0.04	0.04	0.00
C2	587.4	587.4	587.4	75.4	587.4	86.4	587.4	587.4	40.6	0.00	0.00	0.00	0.00
RC2	1000.7	1002.1	1009.8	53.6	1001.9	70.3	1002.3	1009.0	43.6	0.15	0.00	0.03	-0.11
All problems	**973.2**	**974.6**	**977.7**	**47.3**	**974.3**	**59.4**	**974.1**	**977.3**	**45.8**	**0.09**	**-0.04**	**-0.02**	**-0.04**

All values, except for the last line, in Table 5.2 represent the class-based averages. For example, optimal solution of class R1 equals to $(\sum_{i\in R1} optimal\_sol_i)/|R1|$. For all clustered problems (C1 and C2 problem sets) the proposed ALNS finds the optimal solutions. For the other problem sets, our algorithm is able to find relatively good solutions. The average gap between the optimal distances and our best distances is only 0.09% for all problems. The results indicate that on average our algorithm produces better results than both PR and DBL in less time. Our algorithm has also improved the upper bound of the only open Solomon VRPTW instance, i.e. R208. The solution of this problem is provided in Table C.2 of Appendix C.

5.2.2 Results on the Real Numbered Data

Next, we provide results on Solomon's VRPTW benchmark instances with 100 customers, but this time the distances and travel times are not truncated. Since DBL does not present such results, we only compare our algorithm with PR. Table 5.3 contains summary results of this comparison, whereas the detailed results are presented in Table B.2 of Appendix B. The comparisons are again made in terms of the best solution and mean values obtained through 10 runs of each algorithm, where each run lasts for 25000 iterations. The best known solutions were obtained from Yildirim and Catay (2012).

Table 5.3: Summary results of Solomon's real numbered VRPTW instances

	Best Known	PR			The Proposed ALNS			Deviation from (%)		
Class	distance	best dist.	avg. dist.	avg. time (secs)	best dist.	avg. dist.	avg. time (secs)	best known	PR best	PR avg.
R1	1179.63	1209.83	1220.49	53.6	1180.48	1183.41	50.1	0.08	-2.52	-3.19
C1	828.38	828.38	828.38	49.1	828.38	828.38	41.9	0.00	0.00	0.00
RC1	1338.49	1380.91	1391.39	53.38	1338.52	1344.93	49.1	0.00	-2.79	-3.08
R2	877.84	955.42	971.19	150.7	879.50	884.06	42.1	0.18	-7.80	-8.80
C2	589.86	589.86	589.86	87.1	589.86	589.86	39.5	0.00	0.00	0.00
RC2	1004.00	1124.77	1140.06	116.6	1006.36	1013.16	40.7	0.23	-9.82	-10.41
All problems	**977.25**	**1022.27**	**1031.34**	**85.7**	**978.10**	**981.51**	**44.2**	**0.09**	**-3.87**	**-4.34**

For the C1 and C2 problem sets, our algorithm finds the best-known solutions. For the other problem sets, we observe that our algorithm is able to find relatively good solutions. The average gap between the best known distances and our best distances is again only 0.09% for all problems. The results indicate that on average our algorithm produces much better results than PR in almost twice less time. In five instances the best-known solutions are improved: R106, R107, R108, R210 and RC107. The solutions of these problems are provided in Table C.2 of Appendix C.

5.2.3 Analysis of the ALNS Algorithms on VRPTW Instances

In this section, we provide performance results of removal and insertion algorithms based on the results of 56 truncated and 56 real numbered VRPTW instances of Solomon given in the previous sections. We assess the performance of each algorithm by considering statistics on their average usage, required time and ability to generate good solution. The results are presented in Table 5.4. For each algorithm, the "Average Usage" column corresponds to the average usage percentage by running the ALNS heuristic over all truncated and real numbered instances for 10 runs, where each run lasts for 25000 iterations."Average Time" represents the time required per usage and "Average Better Solution" is the average percentage that the corresponding algorithm yields a better solution when it is used throughout the search. The results indicate that although the *RNR* algorithm requires the maximum time among all removal algorithms, it is one of most frequently used removal algorithms and it has the best average better solution performance. Furthermore, the *ZR* algorithm has a quite good "better solution" performance among the least time-consuming removal algorithms. On the other hand, although the *RoR* and the *ZI* algorithms have relatively small better solution performances, the modifications made in this study improved their performance in comparing to the original ones.

Table 5.4: Performance of the ALNS Algorithms on VRPTW instances

Category	Algorithm Abbreviation	Average Usage (%)	Average Time (millisecs.)	Average Better Solution (%)
Removal	RR	9.60	0.02	16.70
Removal	WDR	7.07	0.28	13.60
Removal	WTR	10.22	0.30	20.07
Removal	SR	9.01	0.28	15.41
Removal	PR	7.80	0.23	12.34
Removal	TR	9.91	0.23	18.36
Removal	DR	10.21	0.24	17.50
Removal	HR	2.62	0.55	11.34
Removal	NR	6.16	0.40	16.70
Removal	NNR	4.48	0.02	10.92
Removal	RoR	4.34	0.01	8.01
Removal	ZR	8.48	0.01	16.14
Removal	RNR	10.13	1.25	23.36
Insertion	GI	12.95	1.07	13.40
Insertion	R-2I	24.69	1.68	25.01
Insertion	R-3I	23.44	1.68	23.37
Insertion	R-4I	21.75	1.67	21.11
Insertion	R-mI	14.76	1.61	17.37
Insertion	ZI	0.72	0.32	2.24
Insertion	GIN	1.72	1.83	5.42

The *R-kI* algorithms have a good time-performance balance, as they are frequently chosen due to their better solution performance though they require a lot of time. Average Usage performances of removal and insertion algorithms on VRPTW instances are graphically illustrated in Figure 5.1.

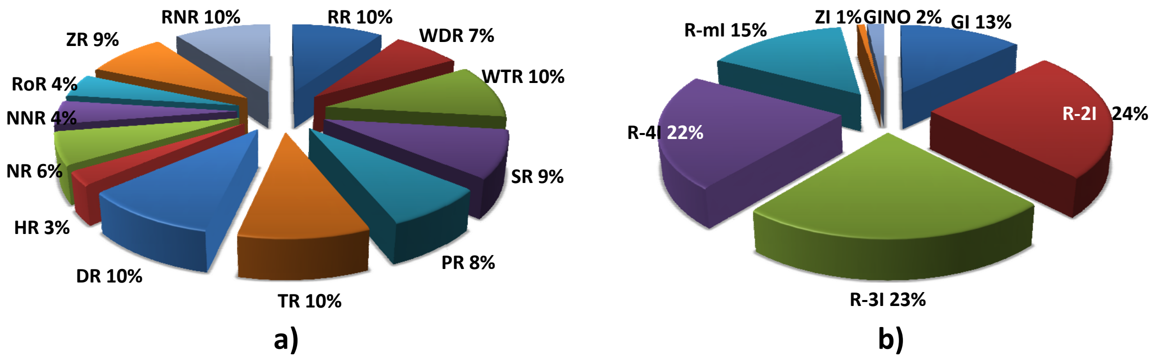


Figure 5.1: Average Usage (%) of **a)** Removal algorithms and **b)** Insertion algorithms on VRPTW instances

5.3 Experiments on VRPVSIPD Instances

In this section, we generate benchmark instances by adapting 25-node, 50-node, and 100-node Solomon problems to VRPVSIPD setting and solve those using the proposed ALNS. For the 25-node instances, we compare our results with those obtained by using CPLEX. For larger instances, we present our results as benchmarks for future studies.

5.3.1 Generation of New Data

There is not much available instance directly related to the considered business model in the existing literature. Therefore, we introduce a new set of benchmark instances based on modifications of the original Solomon VRPTW instances. The original instances come in three sets C, R and RC, classified with respect to the geographical structure of the nodes. We name the modified versions of these sets as M-C, M-R, and M-RC, respectively. Each set consists of two types of problems involving nodes having narrow and wide time window lengths, which are referred to as type 1 and type 2 instances, respectively. Let be the number of nodes, be the service duration, and be the demand of each node of an original instance. To modify an original instance, we first randomly select a certain percentage of nodes to be external vendors. Then, we randomly nominate a certain percentage of the remaining nodes as premium customers. As a result, the remaining nodes are set as regular customers. The time windows of external vendors are set equal to the time windows of the depots. Also, service duration of each external vendor is kept same and considered as the loading time of premium products to vehicles in the modified instances. The time windows associated with the premium customers are adjusted if they lead to any infeasibility. In addition, the size of the premium product set, the volume of each premium product w_p, and a_{vp} values are randomly determined between prespecified lower and upper bounds. During this process, it is assured that each vendor supplies at least one premium product. Furthermore, $B_{|C^p| \times |P|}$ and $Q_{|C^p| \times |P|}$ matrices are also randomly determined between prespecified lower and upper bounds up to a point that the sum of total premium product demand, i.e. $\sum_{p \in P} q_{cp} w_p$, and modified standard product demand of each premium customer equals to the corresponding demand d_i in the original instance.

We have generated our data in 11 different categories for each VRPTW instance set of Solomon, i.e. instances with 25, 50 and 100 customers, by varying p_v, p_{pc}, and $|P|$ values. Category 11 data involving only one premium product is generated for 25-customer case for comparison purposes with CPLEX. Table 5.5 shows the summary of the characteristics of the data categories. Since each instance set contains 56 problems, $(3x10x56 + 1x1x56) = 1736$ new problems are generated in total.

Table 5.5: Key parameters of new benchmark instance categories

Categories		$\boldsymbol{p_v}$ (%)	$\boldsymbol{p_{pc}}$(%)	$\boldsymbol{\lvert P \rvert}$
Category 1	(CTG1)	12.5	20.0	2
Category 2	(CTG2)	12.5	20.0	3
Category 3	(CTG3)	12.5	20.0	4
Category 4	(CTG4)	20.0	$33.\overline{3}$	2
Category 5	(CTG5)	20.0	$33.\overline{3}$	3
Category 6	(CTG6)	20.0	$33.\overline{3}$	4
Category 7	(CTG7)	12.5	$33.\overline{3}$	3
Category 8	(CTG8)	20.0	20.0	3
Category 9	(CTG9)	$16.\overline{6}$	25.0	3
Category 10	(CTG10)	25.0	50.0	3
Category 11(*)	(CTG11)	12.5	20.0	1

(*): Category 11 is only generated for instances with 25 customers

5.3.2 Results on 25-node Instances

We now provide results on the new 25-node instances described in Section 5.3.1 and state some remarks. The proposed ALNS heuristic was applied 10 times to each new 25-node instance and the obtained best known solutions are recorded for future studies. Also, we attempt to solve these instances optimally via CPLEX 12.2 library embedded in a Java platform to evaluate the performance of the ALNS heuristic. First, in order to establish a balance between optimality and feasibility, we set the MIP emphasis parameter of CPLEX to its default value. However, due to the NP-Hardness of the problem, CPLEX was not able to produce feasible solutions with this setting for most of the instances, even with 25 nodes, within 10800 seconds. We then set the MIP emphasis parameter to feasibility in order to emphasize feasibility over optimality. In this way, we have managed to get results for instances with 25 nodes. Table 5.6 presents the performance of both CPLEX and the proposed ALNS heuristic with respect to data categories and problem classes.

Table 5.6: Performance of CPLEX and the proposed ALNS with respect to data categories and problem classes for 25-node instances

	M-C1			M-R1			M-RC1			M-C2			M-R2			M-RC2		
Categories	CPLEX	ALNS	% Gap	CPLEX	ALNS	% Gap	CPLEX	ALNS	% Gap	CPLEX	ALNS	% Gap	CPLEX	ALNS	% Gap	CPLEX	ALNS	% Gap
Category 1	228.5	227.0	-0.60	543.6	528.1	-3.02	516.6	512.2	-0.82	241.3	240.6	-0.30	414.8	388.6	-5.43	423.3	359.5	-12.25
Category 2	259.4	254.2	-1.83	507.4	504.7	-0.59	533.6	533.6	0.00	246.1	245.3	-0.35	415.2	401.5	-3.20	382.7	364.4	-4.06
Category 3	254.7	248.5	-2.53	547.3	543.1	-0.66	573.9	573.9	0.00	246.5	246.3	-0.07	405.4	398.4	-1.72	402.2	381.3	-4.93
Category 4	241.6	236.7	-1.86	509.1	498.4	-2.12	536.3	536.3	0.00	240.0	234.9	-2.02	371.9	362.2	-2.52	377.2	346.1	-7.45
Category 5	260.6	259.3	-0.47	525.2	523.5	-0.29	536.3	536.3	0.00	233.8	233.8	0.00	376.6	370.9	-1.50	379.0	350.2	-6.77
Category 6	277.5	273.7	-1.14	551.4	545.1	-1.17	535.6	535.6	0.00	243.9	238.3	-2.21	383.6	373.4	-2.56	387.3	365.4	-4.99
Category 7	299.7	298.1	-0.53	582.0	563.4	-3.62	546.9	546.9	0.00	255.0	251.7	-1.18	424.9	414.5	-2.44	421.7	383.7	-8.28
Category 8	242.3	238.9	-1.54	487.6	485.7	-0.40	478.4	478.4	0.00	230.2	228.7	-0.59	363.8	359.3	-1.27	354.7	338.6	-4.09
Category 9	271.5	265.1	-2.30	536.7	530.8	-1.21	587.0	587.0	0.00	237.9	236.6	-0.49	383.0	379.0	-1.08	381.1	367.6	-3.56
Category 10	242.0	217.6	-7.68	529.6	527.0	-0.57	-	-	-	260.0	239.0	-6.88	358.6	350.8	-2.16	388.0	345.3	-9.02
Category 11	214.2	214.2	0.00	503.6	498.5	-1.05	405.0	398.6	-1.39	230.7	230.3	-0.18	386.4	378.8	-1.85	336.3	334.5	-0.45
Average	253.8	248.5	-1.86	529.4	522.6	-1.34	525.0	523.9	-0.22	242.3	238.7	-1.30	389.5	379.8	-2.34	384.9	357.9	-5.99

Figure 5.2 provides a more detailed performance of both CPLEX and the proposed ALNS with respect to 11 data categories described in Section 5.3.1. These results indicate that the number of instances solved by CPLEX and the solution quality show a tendency to decrease as the number of premium products $|P|$; vendors $|V|$ and premium customers $|C^P|$ increase.

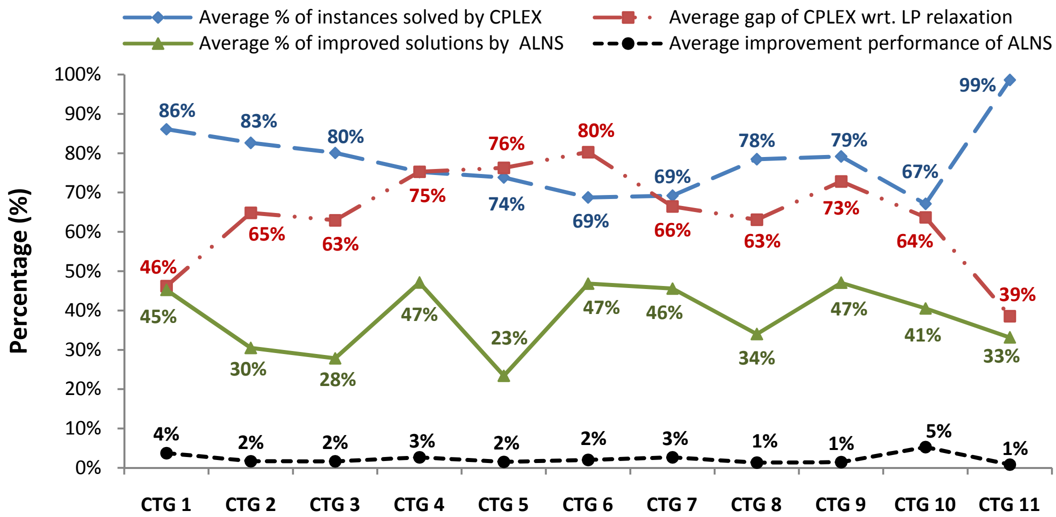


Figure 5.2: Performance of CPLEX and the proposed ALNS with respect to data categories for 25-node instances

CPLEX was able to find only a few optimal solutions within 10800 seconds and none of the results found by the ALNS heuristic, with an average solution time of 6 seconds, for the 25-node instances are worse than the CPLEX results. The results point out that CPLEX is not generally aware of that it has found a good lower limit. The number of improved solutions and the improvement performance of the ALNS heuristic slightly decrease as $|P|$, $|V|$, and $|C^P|$ increase. Figure 5.3 presents a more detailed performance of both CPLEX and the proposed ALNS heuristic with respect to problem class and scheduling horizons of the nodes.

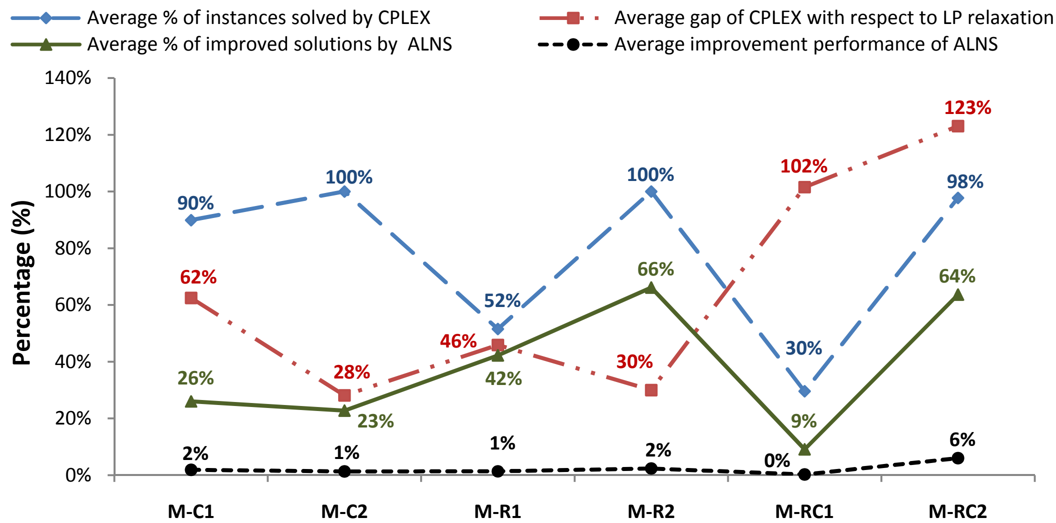


Figure 5.3: Performance of CPLEX and the proposed ALNS with respect to problem classes for 25-node instances

The geographical data (customer and vendor nodes) are random in the new problem sets M-R1 and M-R2, clustered in M-C1 and M-C2, and a mixture of random and clustered in problem sets M-RC1 and M-RC2. Problem sets M-R1, M-C1 and M-RC1 have a short scheduling horizon, whereas the sets M-R2, M-C2 and M-RC2 have a long scheduling horizon. The results in Figure 5.3 indicate that CPLEX had much more difficulty in finding a good feasible solution for M-R1 and M-RC1 instances and through all the data sets it produced the worst lower bounds for M-RC1 and M-RC2 instances. The ALNS heuristic performs significant amount of improvements with respect to the best known CPLEX solutions of M-R1, M-R2 and M-RC2 instances. The detailed heuristic and CPLEX results of new instances with 25 nodes are available in Appendix D.

5.3.3 Results on 50- and 100-node Instances

Unlike 25-node instances, we can only provide detailed heuristic results for instances with 50 nodes and 100 nodes, since CPLEX was not able to produce any feasible results on these instances within the specified time limit even the MIP emphasis parameter is set to feasibility. The proposed ALNS heuristic was applied 10 times to each new 50- and 100-node instance and the obtained best known solutions are recorded for future studies. Table 5.7 presents summary results of these best known solutions with respect to data categories and problem classes. Detailed heuristic results of 50- and 100-node instances are provided in Appendix E and Appendix F, respectively.

Table 5.7: Summary results of the proposed ALNS with respect to data categories and problem classes for 50- and 100-node instances

	50 Node-Problems						100 Node-Problems					
Categories	**M-C1**	**M-R1**	**M-RC1**	**M-C2**	**M-R2**	**M-RC2**	**M-C1**	**M-R1**	**M-RC1**	**M-C2**	**M-R2**	**M-RC2**
Category 1	468.9	829.7	911.9	386.1	581.7	630.9	1026.6	1208.1	1372.2	618.6	840.1	992.6
Category 2	496.2	892.9	1022.8	388.2	587.4	647.9	1076.0	1241.8	1413.5	631.9	851.6	1001.0
Category 3	559.8	877.1	983.4	399.4	596.7	662.8	1113.9	1254.1	1476.9	643.6	857.7	1012.7
Category 4	452.3	758.5	794.9	369.6	555.4	587.5	999.6	1103.2	1234.1	604.9	796.4	923.5
Category 5	485.0	785.4	935.0	379.0	566.7	621.4	1032.0	1133.9	1273.6	605.7	802.2	949.0
Category 6	529.7	816.1	967.8	387.6	580.5	629.7	1089.9	1149.5	1326.8	630.0	804.3	955.8
Category 7	574.0	937.0	1203.3	398.5	601.5	665.4	1151.1	1284.9	1479.6	651.5	862.8	1006.8
Category 8	441.1	748.3	817.2	363.6	556.3	603.8	958.1	1083.9	1219.4	591.3	792.9	915.5
Category 9	454.0	789.9	871.2	371.2	570.6	607.3	1021.9	1131.7	1298.2	622.4	820.9	966.1
Category 10	458.4	771.9	921.9	369.2	543.2	583.0	1001.7	1060.7	1193.6	592.6	780.6	913.0
Average	491.9	820.7	942.9	381.2	574.0	624.0	1047.1	1165.2	1328.8	619.3	821.0	963.6

5.3.4 Analysis of the ALNS Algorithms on VRPVSIPD Instances

In this section, we provide average performance results of removal, vendor selection/allocation and insertion algorithms based on the results of 100-node new benchmark instances given in the previous sections. The average performance results are presented in Table 5.8, with the same definitions in Section 5.2.3.

Table 5.8: Performance of the ALNS Algorithms on VRPVSIPD instances

Category	Algorithm Abbreviation	Average Usage (%)	Average Time (millisecs.)	Average Better Solution (%)
Removal	RR	7.81	0.03	7.93
Removal	WDR	6.89	0.25	8.31
Removal	WTR	9.88	0.26	9.02
Removal	SR	9.05	0.23	7.67
Removal	PR	7.47	0.20	6.33
Removal	TR	10.14	0.19	9.01
Removal	DR	9.76	0.21	8.28
Removal	HR	2.66	0.45	8.81
Removal	NR	6.36	0.34	10.70
Removal	NNR	4.58	0.03	8.26
Removal	RoR	6.19	0.01	8.40
Removal	ZR	8.45	0.03	9.06
Removal	RNR	10.33	0.82	12.49
Insertion	GI	11.59	2.89	7.26
Insertion	R-2I	25.29	3.73	13.33
Insertion	R-3I	23.91	3.66	12.70
Insertion	R-4I	21.83	3.72	11.74
Insertion	R-mI	14.63	3.89	10.19
Insertion	ZI	0.61	1.02	1.61
Insertion	GIN	2.15	4.57	4.70
Vendor Selection	NNVS	8.75	0.01	7.91
Vendor Selection	RNVS	39.45	0.15	14.78
Vendor Selection	NNVSN	1.25	0.01	2.14
Vendor Selection	RNVSN	38.00	0.15	14.77
Vendor Selection	RVS	2.48	0.01	3.56
Vendor Selection	HVS	9.08	0.01	9.59

We observe that the average better solution performance of each removal and insertion algorithm is decreased comparing to the results presented in Section 5.2.3. Individual adaptive scoring of more than two algorithm categories may have yielded such a result. Therefore, considering a new adaptive scoring mechanism directly based on the combinations of removal, vendor selection/allocation and insertion algorithms may improve these results. The *RNR* is the most frequently used removal algorithm and has the best average better solution performance, although it requires the maximum time among all removal algorithms. Furthermore, the *RoR* is the fastest removal algorithm and has the best time-better solution balance, whereas the *ZR* algorithm has an average performance. On the other hand, the *ZI* algorithm required the least average time but it is not used frequently as it rarely yields better solutions. The *R-kI* algorithms have again good time-performance balance, as they are frequently chosen due to their better solution performance though they require a lot of time. Among all vendor selection/allocation algorithms, the *RNVS* and *RNVSN* mechanisms are clearly the most preferred ones with almost twice the better performance, although they are the most time consuming vendor selection/allocation algorithms. Average usage performances of removal, vendor selection/allocation and insertion algorithms on VRPVSIPD instances are graphically illustrated in Figure 5.4.

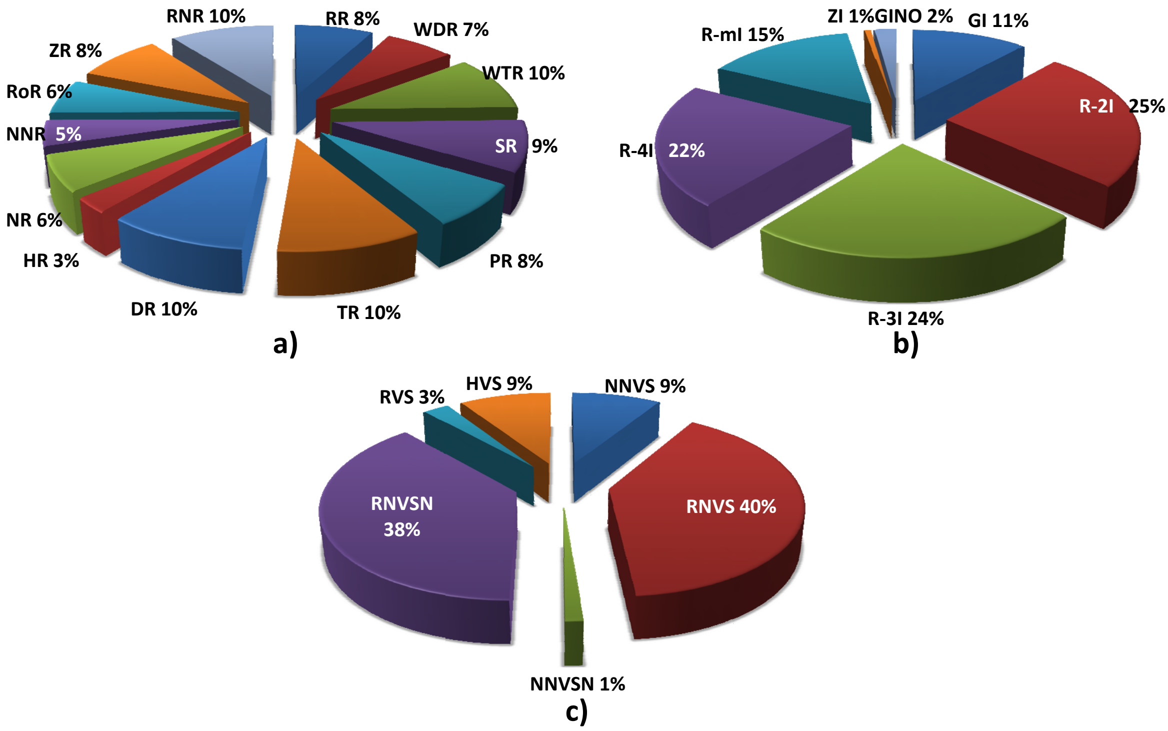


Figure 5.4: Average Usage (%) of **a)** Removal algorithms, **b)** Insertion algorithms, and **c)** Vendor selection/allocation algorithms on VRPVSIPD instances.

Furthermore, average usage performances on Category 6 benchmark instances of the ALNS algorithms with respect to data categories, which are determined by geographical structure and time window lengths of nodes, are given in Table 5.9. This analysis requires considerable amount of time, so we only considered Category 6 instances for this analysis since they represent the main idea of VRPVSIPD quite well as they have a fair balance among the number of premium products, external vendors, and premium customers. The results indicate that the adaptive preferences of removal, vendor selection/allocation, and insertion algorithms do not significantly depend on geographical structures of nodes. On the other hand, the average usages of *RR*, *RoR*, *GI*, and *R-kI* significantly change with respect to time window lengths of nodes. Since 2XX problems have wider time windows, inserting a removed customer is much easier comparing to 1XX problems. So, obtaining a better feasible solution from a partial solution requires much effort in 1XX problems. Consequently, the *RR* and *GI* are used more frequently in 2XX problems, whereas the *RoR* is used more frequently in 1XX problems mainly because of this reason. Although the *R-mI* is used more frequently and therefore the remaining regret insertion algorithms are preferred less in 2XX problems comparing to 1XX problems, it is not directly because of the remarkable performance of the *R-mI*. Since there are less than five routes in 2XX problems, when the *R-mI* algorithm is selected it actually behaves like one of the *R-kI* algorithms where $k < 5$. In other words, the *R-mI* includes some partial adaptive scores of the *R-2I*, *R-3I*, and*R-4I* algorithms. So, we can conclude that performance of the *R-mI* decreases as "*m*" increases.

Table 5.9: Average Usage (%) performances of the ALNS Algorithms with respect to Data Categories on Category 6 benchmark instances

	Algorithm	Average Usage (%) with respect to Data Categories					
Category	Abbreviation	M-C1	M-R1	M-RC1	M-C2	M-R2	M-RC2
Removal	RR	6.70	6.73	7.21	10.01	9.34	8.89
Removal	WDR	5.85	6.07	6.87	8.64	7.10	7.45
Removal	WTR	9.01	9.16	10.32	8.70	10.19	10.19
Removal	SR	10.94	8.39	7.64	9.08	10.05	9.69
Removal	PR	6.70	6.32	6.47	9.68	7.70	8.74
Removal	TR	10.72	11.19	10.79	8.99	10.15	9.95
Removal	DR	8.19	9.24	8.83	10.02	10.18	10.74
Removal	HR	1.78	2.56	1.91	4.46	2.68	2.29
Removal	NR	8.13	5.42	6.55	6.90	5.39	5.62
Removal	NNR	3.90	4.15	3.51	5.94	5.32	4.57
Removal	RoR	9.85	10.61	10.71	3.01	1.77	2.05
Removal	ZR	7.43	8.35	7.62	8.83	9.55	8.93
Removal	RNR	10.79	11.81	11.57	5.74	10.57	10.91
Insertion	GI	10.18	8.84	9.03	14.42	13.32	14.23
Insertion	R-2I	27.07	29.55	28.92	21.63	21.89	22.26
Insertion	R-3I	25.00	26.93	26.89	20.99	21.06	21.15
Insertion	R-4I	22.81	22.87	22.13	20.80	20.96	20.69
Insertion	R-mI	12.21	8.02	8.81	20.70	20.69	20.05
Insertion	ZI	0.57	0.95	1.39	0.26	0.23	0.29
Insertion	GIN	2.17	2.85	2.82	1.20	1.84	1.33
Vendor Selection	NNVS	9.85	8.03	6.55	2.60	5.70	4.93
Vendor Selection	RNVS	39.85	40.74	41.15	44.85	42.47	44.92
Vendor Selection	NNVSN	0.85	0.87	0.85	0.43	1.35	0.85
Vendor Selection	RNVSN	39.78	39.97	40.98	44.64	42.20	44.89
Vendor Selection	RVS	1.39	1.67	1.96	0.85	2.08	1.41
Vendor Selection	HVS	8.28	8.71	8.52	6.63	6.19	3.01

Chapter 6

Case Study

In this chapter, a case study based on a real dataset in the city of Istanbul, Turkey is presented under two different scenarios; 1) serving customers within classic time windows (a variant of the traditional VRPTW), and 2) serving customers within predefined time slots (a variant of the CVRP with time length restrictions). In both scenarios, premium products are collected from multiple vendor source locations and then delivered to customer locations in a single basket with standard products. The proposed solution methodology in Chapter 4 is integrated with the ArcGIS 10.0 development environment to provide a convenient user interface on a geographical information system. A cross-sectional image of this interface is given in Figure 6.1. Network data, premium products list, depots, vendors, regular customers, premium customers and vehicle capacity are all entered as a layer or a data input via this interface. After that, the user specifies the time window handling process, i.e. the scenario type, and chooses how to visualize the obtained routes. Finally, the user solves the problem for the specified scenarios and gets the analysis results that include the total route distance, route duration, number of vehicles required, vehicle utilization, and average number of vendors visited.

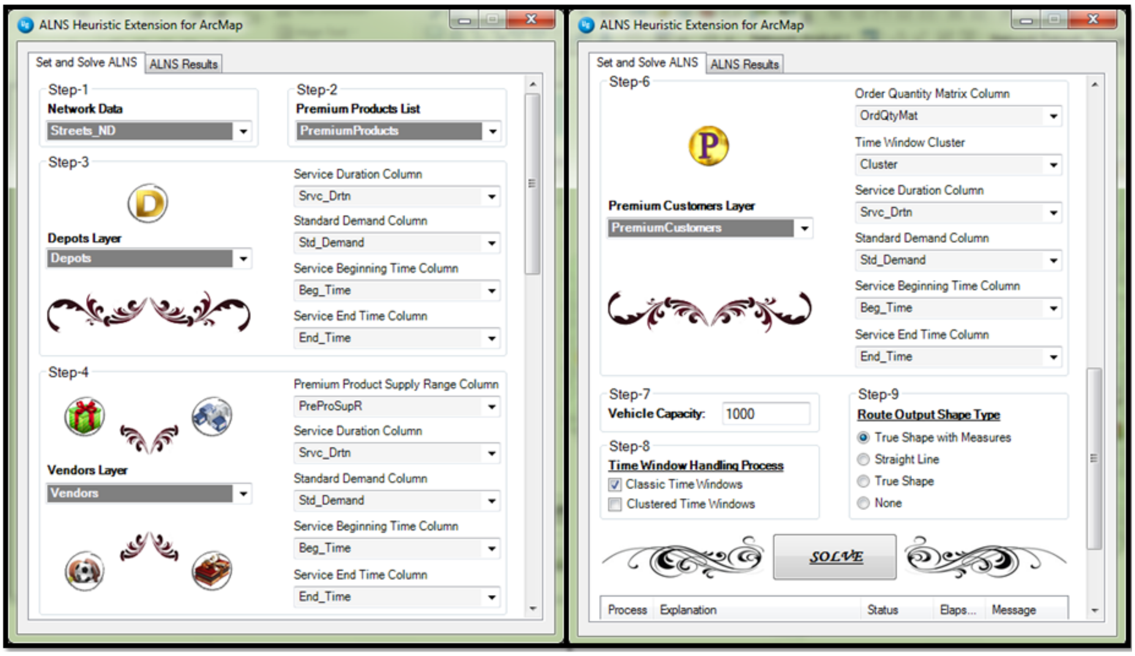


Figure 6.1: A cross-sectional image of the ArcGIS-based interface

The study area covers the surroundings of an online retailer in the Ataşehir district located on the Asian side of Istanbul. We consider three different premium products, namely books, souvenirs, and sports equipment. In Figure 6.2, the individual suppliers of these products are represented by book, gift box, and sports equipment symbols, respectively. Also in Figure 6.2, we have "*super vendors*" that supply more than one premium product at the same time. These super vendors are represented by super market symbols. Online retailer store (depot), premium customers and regular customers are shown with capital letter symbols D, P, and R, respectively. We have one depot, four super vendors, thirteen individual vendors, fifty-two regular customers, and twenty-three premium customers in this case study.

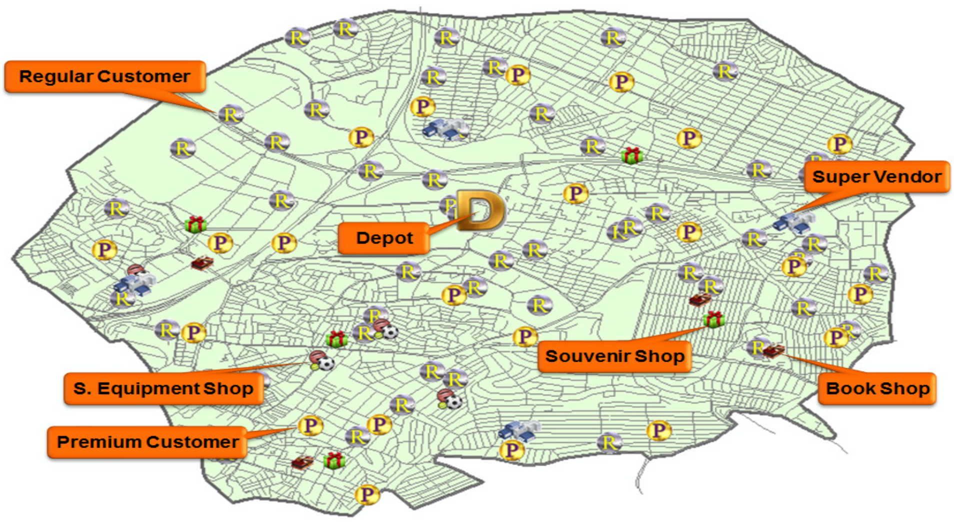
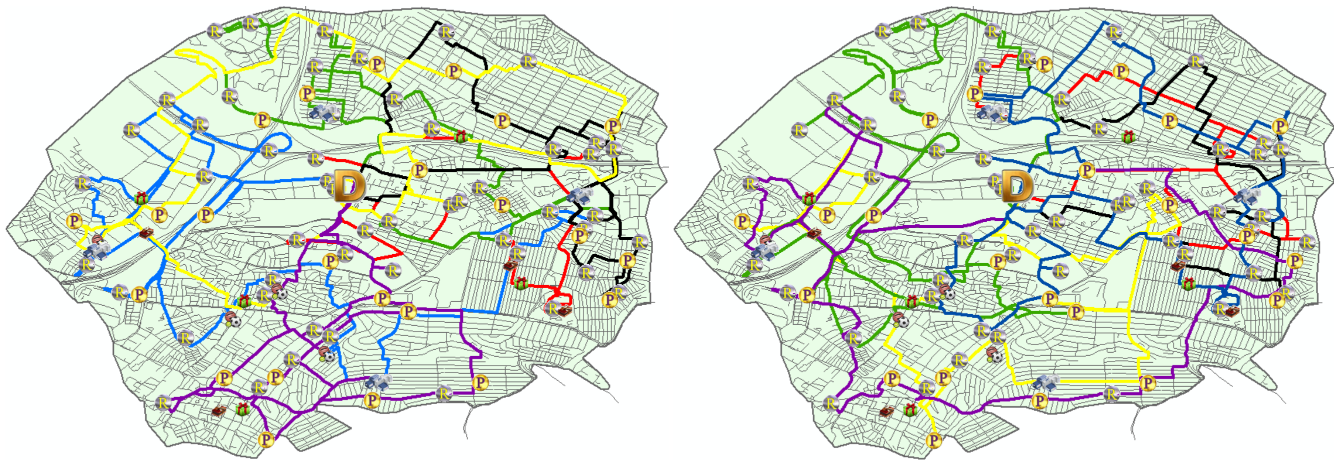


Figure 6.2: Problem setting in Ataşehir, Istanbul

In the first scenario, we solve the VRPVLIPD with traditional time windows by serving both premium and regular customers within their individual and independent time windows in a nine-hour time horizon. In the second scenario, we consider three consecutive time slots, which are three hours in length. In other words, the deliveries are grouped into specific time slots based on the time window preferences of customers. Thus, the deliveries in the same time slot may be treated as orders with no time windows. Compared to the first scenario, the second one is a less flexible approach but it reduces the problem size dramatically as a result of time-based clustering. In both scenarios, a number of vehicles leave the store with all standard products loaded and stop by some vendors, if necessary, on their way to visit customers sequentially to make deliveries. We present in Figure 6.3, the routing solutions obtained by running the proposed ALNS heuristic under Scenario 1 and Scenario 2 respectively.



(a) Scenario 1	**Key Factors**	**(b) Scenario 2**
250	Solution Time (sec.)	125
6	Required # of Vehicles	2
178	Total Distance (km)	156
40	Total Duration (h)	15
92%	Average Vehicle Capacity Usage (%)	90%

Figure 6.3: Results for the case study with Scenario 1 and Scenario 2

We assume a specific number of time slots in the analysis illustrated in Figure 6.3. However, the number of time slots may not be foreseen in advance. In order to investigate and evaluate the effect of time slots, we conduct a sensitivity analysis in Table 6.1. This analysis exhibits that the required number of vehicles and solution time shows a concave relationship with the number of time slots. Furthermore, the total distance and total duration decrease as the number of time slots decreases, whereas the average vehicle capacity usage stays almost stationary with respect to the number of time slots.

Table 6.1: Sensitivity analysis results with respect to number of time slots

	Number of Time Slots					
Key Factors	**No Time Slot**	**7**	**5**	**3**	**2**	**1**
Duration per Time Slot (h)	-	1.3	1.8	3	4.5	9
Solution Time (sec.)	250	235	205	125	150	180
Required # of Vehicles	6	4	3	2	3	5
Total Distance (km)	178	172	161	156	142	120
Total Duration (h)	40	30	24	15	14	13
Average Vehicle Capacity Usage (%)	92	85	90	90	93	97

Chapter 7

Conclusion and Future Research

In this study, we present an alternative model to make distribution planning for the collaborative relationship between online retailers and external vendors, generate a new set of benchmark instances, develop an efficient solution procedure based on the ALNS framework, conduct an extensive computational study to test its performance, and perform a case study. The proposed ALNS heuristic uses new as well as existing removal and insertion algorithms, which improve the solution quality. Also, it introduces new algorithms for vendor selection/allocation operations. We believe these algorithms can also be used in similar ALNS frameworks for solving other types of problems. Our tests on both the new benchmark instances and Solomon's VRPTW instances show that the proposed ALNS heuristic is capable of obtaining high quality solutions in reasonable amounts of time. Moreover, we observe that the proposed ALNS heuristic, equipped with new removal and insertion algorithms, has been able to discover new best solutions to both truncated and real numbered VRPTW benchmark instances of Solomon. On the other hand, tests on new benchmark instances indicate that the VRPVSIPD problem gets more difficult as the number of premium products and the percentage of vendors as well as premium customers are increased. Additionally, sensitivity analysis on the case study puts forward that serving customers within proper amount of time slots makes more sense with respect to total distance, total duration and required number of vehicles.

Further research on this topic may focus on developing more specific and efficient removal as well as insertion algorithms for only external vendors, i.e. algorithms for just removing existing vendors from a route and inserting a new proper set of vendors into the route without removing any premium customers. In addition, both the mathematical model and the proposed ALNS heuristic may be adapted to the multi-depot case. Also, there may be capacity limit for depots and vendors in which case vendors may supply standard products up to a certain limit. Since vendors can be treated as (sub) depots in this scenario, the VRPVSIPD problem can be considered as a variant of VRP with intermediate facilities/satellites. Multiple pick-ups and multiple deliveries process may be introduced for both regular and premium customers. More specific worst removal cost functions that also take into account the cost of allocated vendors, and new specific Shaw removal relatedness measures can be introduced for premium customers. New adaptive scoring mechanisms directly based on the combinations of removal, vendor selection/allocation and insertion algorithms may be taken into account. Vehicles may be allowed to make multiple trips and the minimization of number of vehicles as a secondary objective can be integrated into the process.

Bibliography

Angelelli, E. and Speranza, M.G. (2002). The periodic vehicle routing problem with intermediate facilities. *European Journal of Operational Research*, 137, 233-247.

Attanasio, A., Cordeau, J. F., Ghianic, G., and Laporte, G. (2004). Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. *Parallel Computing*, 30(3), 377–387.

Bard, J. F., Huang, L., Dror, M., and Jaillet, P. (1998a). A branch and cut algorithm for the VRP with satellite facilities. *IIE Transactions*, 30 (9), 821-834.

Bard, J. F., Huang, L., Dror, M., and Jaillet, P. (1998b). A decomposition approach to the inventory routing problem with satellite facilities. *Transportation Sci*., 32, 189-203.

Bent, R. and van Hentenryck, P. (2006). A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. *Computers & Operations Research*, 33(4), 875–893.

Berbeglia, G., Cordeau, J. F., Gribkovskaia, I., and Laporte, G. (2007). Static pickup and delivery problems, a classification scheme and survey, *TOP*, 15, 1, 1-31.

Berbeglia, G., Cordeau, J. F., and Laporte, G. (2010). Dynamic pickup and delivery problems. *European Journal of Operational Research*, 202 (1), 8–15.

Bodin, L. D. and Sexton, T. (1986). The multi-vehicle subscriber dial-a-ride problem. *TIMS Studies in the Management Sciences*, 22, 73–86.

Borndörfer, R., Klostermeier, F., Grötschel, M., and Küttner, C. (1997). Telebus Berlin, vehicle scheduling in a dial-a-ride system, Technical report SC 97–23, *Konrad-Zuse-ZentrumfürInformationstechnik*, Berlin.

Bozkaya, B., De Kervenoael, R., and Aykac, D. S. O. (2009). Premium e-grocery: Exploring value in logistics integrated service solutions. *The International Conference for Prospects for Research in Transport and Logistics on a Regional-Global Perspective*, February 12-14, Istanbul.

Caricato, P., Ghiani, G., Grieco, A., and Guerriero, E. (2003). Parallel tabu search for a pickup and delivery problem under track contention. *Parallel Computing*, 29, 631–639.

Cordeau, J. F. and Laporte, G. (2003a). A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research B*, 37, 579–594.

Cordeau, J. F. and Laporte, G. (2003b). The dial-a-ride problem (DARP), variants, modeling issues and algorithms. *4OR, A Quarterly Journal of Operations Research*, 1, 89–101.

Cordeau, J. F. and Laporte, G. (2007). The dial-a-ride problem, models and algorithms. *Annuals of Operations Research*, 153, 29–46.

Cordeau, J. F., Laporte, G., and Ropke, S. (2008). Recent models and algorithms for one-to-one pickup and delivery problems. In: *The Vehicle Routing Problem: Latest Advances and New Challenges* Vol. 43, 327–357.

Cordeau, J. F., Laporte, G., Pasin, F., and Ropke, S. (2010). Scheduling technicians and tasks in a telecommunications company. *Journal of Scheduling*, 13, 393-409.

Cullen, F. H., Jarvis, J. J., and Ratliff, H. D. (1981). Set partitioning based heuristics for interactive routing. *Networks*, 11(2), 125–143.

Demir, E., Bektas, T., and Laporte, G. (2012). An adaptive large neighborhood search heuristic for the pollution-routing problem. *European Journal of Operational Research*, 223, 346-359.

Desaulniers, G., Desrosiers, J., Erdmann, A., Solomon, M. M., and Soumis, F. (2002).VRP with Pickup and Delivery. In: *The Vehicle Routing Problem*, *edsToth*, P. and Vigo, D., 225-237, Philadelphia, SIAM.

Desrosiers, J., Dumas, Y., Soumis, F., Taillefer, S., and Villeneuve, D. (1991). An algorithm for mini-clustering in handicapped transport. *Les Cahiers du GERAD*, G-91-02, HEC Montréal.

Doerner, K., Hartl, R. F., and Reimann, M. (2001). Ants solve time constrained pickup and delivery problems with full truckloads. In: *Operations Research Proceedings*, 395–400.

Doerner, K., Hartl, R. F., Maniezzo, V., and Reimann, M. (2003). An ant system metaheuristic for the capacitated arc routing problem. *Proceedings of the 5th metaheuristics international conference*, Tokyo, Japan.

Dumas, Y., Desrosiers, J., and Soumis, F. (1989). Large scale multi-vehicle dial-a-ride problems. *Les Cahiers du GERAD*, G-89-30, HEC Montréal.

Ganesh, K. and Narendran, T. T. (2007). CLOVES: A cluster-and-search heuristic to solve the vehicle routing problem with delivery and pick-up. *European Journal of Operational Research*, 178(3), 699–717.

Gendreau, M., Potvin, J. Y., Bräumlaysy, O., Hasle, G., and Løkketangen, A. (2007). Metaheuristics for the vehicle routing problem and its extensions, a categorized bibliography. In: *The Vehicle Routing Problem: Latest Advances and New Challenges* Vol. 43, 143–169.

Hart, S. M. (1996). The modeling and solution of a class of dial-a-ride problems using simulated annealing. *Control and Cybernetics*, 25 (1), 131–157.

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

Kristiansen, S., Sørensen, M., Stidsen, T., and Herold, M. (2013). The consultation timetabling problem at Danish high schools. *Journal of Heuristics*, to appear.

Li, H. and Lim, A. (2001). A metaheuristic for the pickup and delivery problem with time windows. In: *13th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'01)*, IEEE Computer Society, Los Alamitos, CA, 333–340.

Lin, C. K. Y. (2008). A cooperative strategy for a vehicle routing problem with pickup and delivery time windows. *Computers and Industrial Engineering*, 55, 766–782.

Liu, J., Liu, D., Liu, M., and He, Y. (2010). An improved multiple ant colony system for the collection vehicle routing problems with intermediate facilities. *Proceedings of the 8th World Congress on Intelligent Control and Automation*, Jinan, China.

Masson, R., Lehuédé, F., and Péton, O. (2012). An adaptive large neighborhood search for the pickup and delivery problem with transfers. *Transportation Science*, articles in advance, doi:10.1287/trsc.1120.0432.

Melachrinoudis, E., Ilhan, A. B., and Min, H. (2007). A dial-a-ride problem for client transportation in a healthcare organization. *Computers & Operations Research*, 34, 742–759.

Montane, F. A. and Galvao, 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.

Muller, L. F. (2009). An adaptive large neighborhood search algorithm for the resource-constrained project scheduling problem. In *Proceedings of the VIII Metaheuristics International Conference (MIC) 2009*, Hamburg, Germany.

Muller, L. F., Spoorendonk, S., and Pisinger, D. (2012). A hybrid adaptive large neighborhood search heuristic for lot-sizing with setup times. *European Journal of Operational Research*, 218(3), 614-623.

Nagy, G. and 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.

Nanry, W. P. and Barnes, W. (2000). Solving the pickup and delivery problem with time windows using reactive tabu search. *Transport Research Part B*, 34, 107–121.

Parragh, S. N., Doerner, K. F., and Hartl, R. F. (2008a). A survey on pickup and delivery problems Part I, Transportation between customers and depot. *Journal fürBetriebswirtschaft*, 58, 1, 21-51.

Parragh, S. N., Doerner, K. F., and Hartl, R. F. (2008b). A survey on pickup and delivery problems Part II, Transportation between pickup and delivery locations. *Journal fürBetriebswirtschaft*, 58, 1, 81-117.

Pisinger, D. and Ropke, S. (2007). A general heuristic for vehicle routing problems. *Comput. Oper. Res.*, 34 (8), 2403–2435.

Pisinger, D. and Ropke, S. (2010). Large neighborhood search. In: Gendreau M, Potvin J-Y, editors, *Handbook of Metaheuristics of International Series in Operations Research & Management Science*, vol. 146, 399–419.

Polacek, M., Doerner, K. F., Hartl, R. F., and Maniezzo, V. (2008). A variable neighborhood search for the capacitated arc routing problem with intermediate facilities. *J Heuristics*, 14(5), 405-423.

Rekiek, B., Delchambre, A., and Saleh, H. A. (2006). Handicapped person transportation, an application of the grouping genetic algorithm. *Engineering Application of Artificial Intelligence*, 19, 511–520.

Ribeiro, G. M. and Laporte, G. (2012). An Adaptive Large Neighborhood Search Heuristic for the Cumulative Capacitated Vehicle Routing Problem. *Computers & Operations Research*, 39 (3), 728-735.

Roberti, R. (2012). Exact Algorithms for Different Classes of Vehicle Routing Problems. *PhD in Control System Engineering and Operational Research*, *University of Bologna*.

Ropke, S. and Pisinger, D. (2006a). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science,* 40 (4), 455–472.

Ropke, S. and Pisinger, D. (2006b). A unified heuristic for a large class of vehicle routing problems with backhauls. *European Journal of Operational Research*, 171 (3), 750–775.

Sevilla, F. C. and de Blas, C. S. (2003). Vehicle routing problem with time windows and intermediate facilities. *S.E.I.O.'03 Edicions de la Universitat de Lleida*, 3088-3096.

Shaw, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. *Proceedings of the 4th International Conference on Principles and Practice of Constraint Programming*. Springer, New York, 417–431.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time windows. *Operations Research*, 35:254-265.

Tarantilis, C. D., Zachariadis, E. E., and Kiranoudis, C.T. (2008). A Hybrid Guided Local Search for the Vehicle-Routing Problem with Intermediate Replenishment Facilities. *International Journal on Computing*, 20(1), 154-168.

Thangiah, S. R., Fergany, A., and Awan, S. (2007). Real-time split-delivery pickup and delivery time window problems with transfers. *Central European Journal of Operations Research*, 15(4), 329–349.

Toth, P. and Vigo, D. (1996). Fast local search algorithms for the handicapped persons transportation problem. In: *Metaheuristics: Theory and Applications*, Osman, I.H. and J.P. Kelly (eds.), Boston, Kluwer, 677–690.

Toth, P. and Vigo, D. (1997). Heuristic algorithms for the handicapped persons transportation problem. *Transportation Science*, 31, 60–71.

Toth, P. and Vigo, D. (2002a). An overview of vehicle routing problems. In: Toth P., Vigo D. (eds) *The Vehicle Routing Problem*. *SIAM Monographs on Discrete Mathematics and Applications*. vol. 9. SIAM, Philadelphia, 1–26.

Toth, P. and Vigo, D. (2002b). VRP with backhauls. In: Toth P, Vigo D (eds) *The Vehicle Routing Problem*. *SIAM Monographs on Discrete Mathematics and Applications*. vol. 9. SIAM, Philadelphia, 195–224.

Yanik, S., Bozkaya, B., and De Kervenoael, R. (2013). A New VRPPD Model and a Hybrid Heuristic Solution Approach for E-Tailing. *European Journal of Operational Research*, forthcoming.

Yildirim, U. M. and Çatay, B. (2012). A time-based pheromone approach for the ant system. *Optimization Letters*, 6(6), 1081–1099.

Appendix A: Parameter tuning summary results

We report here the initial setting, tuning sequence, range and deviation of each parameter used in the proposed ALNS heuristic. In Table A.1, the first value of each parameter is its initial setting.

Table A.1: Parameter tuning summary results.

Parameter	Tuning Sequence	Parameter Settings and Corresponding Deviations										
σ_2	1	Value	9	0	2	4	6	12	14	**16***	18	20
		Deviation	0.87%	0.93%	0.96%	0.99%	0.91%	0.99%	0.87%	**0.73%**	0.80%	0.95%
N_{SI}	2	Value	**100***	50	150	200	250	300	350	400	450	500
		Deviation	**0.73%**	0.74%	0.91%	0.94%	0.95%	0.88%	0.94%	0.86%	0.89%	0.91%
N_{IwI}	3	Value	25000	100	500	1000	2000	3000	**4000***	5000	6000	10000
		Deviation	0.74%	1.63%	0.98%	0.89%	0.87%	0.96%	**0.73%**	0.75%	0.78%	0.75%
ρ	4	Value	**0.1***	0.05	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
		Deviation	**0.73%**	0.78%	0.92%	0.79%	0.88%	0.78%	0.88%	0.92%	0.81%	0.82%
σ_1	5	Value	33	5	10	**20***	25	30	35	40	45	50
		Deviation	0.73%	0.85%	0.90%	**0.71%**	0.88%	0.89%	1.01%	0.86%	0.96%	0.79%
σ_3	6	Value	**13***	3	6	9	12	15	21	24	27	30
		Deviation	**0.71%**	0.81%	0.86%	0.85%	0.83%	0.81%	0.77%	0.88%	0.81%	0.93%
ϕ_1	7	Value	**9***	0.5	1	3	5	7	11	13	15	
		Deviation	**0.71%**	0.77%	0.93%	0.93%	0.86%	0.87%	0.86%	0.96%	0.87%	
ϕ_2	8	Value	**3***	0.25	1	5	7	9	11	13	15	
		Deviation	**0.71%**	0.73%	0.84%	0.87%	0.90%	0.81%	0.73%	0.87%	0.90%	
ϕ_3	9	Value	**5***	0.15	1	3	7	9	11	13	15	
		Deviation	**0.71%**	0.86%	0.79%	0.91%	0.86%	0.84%	0.87%	0.85%	0.77%	
ϕ_4	10	Value	**2***	0.25	1	3	4	5	6	7	8	9
		Deviation	**0.71%**	0.85%	0.82%	0.77%	0.76%	0.88%	0.88%	0.84%	0.81%	0.80%
μ	11	Value	**0.05***	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
		Deviation	**0.71%**	0.92%	0.87%	0.97%	0.98%	1.01%	1.11%	1.12%	1.10%	1.30%
ε	12	Value	**0.9998***	0.999	0.9991	0.9992	0.9993	0.9994	0.9995	0.9996	0.9997	0.9999
		Deviation	**0.71%**	1.37%	1.12%	1.26%	1.09%	1.08%	1.04%	0.89%	0.84%	0.95%
	13	Value	**0.025***	0.05	0.075	0.1	0.125	0.15	0.175	0.2		
		Deviation	**0.71%**	0.90%	1.04%	0.98%	0.90%	0.90%	0.89%	0.96%		
κ	14	Value	**3***	1	2	4	5	6				
		Deviation	**0.71%**	0.85%	0.84%	0.85%	0.82%	0.89%				
η	15	Value	**6***	2	4	8	10	12				
		Deviation	**0.71%**	0.75%	0.82%	0.75%	0.82%	0.83%				
γ	16	Value	**0.2***	0	0.05	0.1	0.15	0.25				
		Deviation	**0.71%**	0.92%	0.81%	0.88%	0.87%	0.75%				
ξ	17	Value	**2***	0	1	3	4	5				
		Deviation	**0.71%**	0.78%	0.76%	0.74%	0.88%	0.84%				
n_Z	18	Value	**11***	5	7	9	13	15	19	21	25	30
		Deviation	**0.71%**	0.73%	0.78%	0.87%	0.82%	0.75%	0.77%	0.73%	0.80%	0.86%
ψ	19	Value	0.25	0.05	0.1	0.15	0.2	0.3	**0.35***	0.4	0.45	0.5
		Deviation	0.10%	0.12%	0.08%	0.12%	0.08%	0.08%	**0.04%**	0.08%	0.08%	0.09%

Appendix B: Truncated and Real numbered ALNS results of Solomon's VRPTW instances

We report here the truncated and real numbered ALNS heuristic results of Solomon's VRPTW benchmark instances.

Table B.1: Truncated numbered ALNS results of Solomon's VRPTW instances.

Truncated numbered results of Solomon's VRPTW instances

	Optimal	PR			DBL		The Proposed ALNS			Deviation from (%)			
Instance	distance	best dist.	avg. dist.	avg. time (secs)	best dist.	avg. time (secs)	best dist.	avg. dist.	avg. time (secs)	optimal	PR best	DBL best	PR avg.
R101	1637.7	1637.7	1638.6	30.0	1637.7	48.0	**1637.7**	1639.0	62.5	0.00	0.00	0.00	0.02
R102	1466.6	1467.7	1467.7	33.0	1466.6	46.0	**1466.6**	1467.4	60.6	0.00	-0.07	0.00	-0.02
R103	1208.7	1208.7	1208.9	34.0	1208.7	46.0	**1208.7**	1209.3	52.2	0.00	0.00	0.00	0.03
R104	971.5	976.0	977.1	34.0	971.5	45.0	976.0	977.8	45.4	0.46	0.00	0.46	0.07
R105	1355.3	1355.3	1355.8	31.0	1355.3	47.0	**1355.3**	1356.6	53.2	0.00	0.00	0.00	0.06
R106	1234.6	1234.6	1234.6	33.0	1234.6	46.0	**1234.6**	1235.2	50.3	0.00	0.00	0.00	0.05
R107	1064.6	1064.6	1068.2	33.0	1064.6	42.0	**1064.6**	1068.0	47.9	0.00	0.00	0.00	-0.02
R108	932.1	933.7	943.5	36.0	936.1	44.0	**932.1**	939.4	45.6	0.00	-0.17	-0.43	-0.43
R109	1146.9	1146.9	1150.2	31.0	1146.9	46.0	**1146.9**	1148.4	50.0	0.00	0.00	0.00	-0.16
R110	1068.0	1075.6	1083.1	33.0	1073.9	49.0	**1068.0**	1079.8	50.4	0.00	-0.71	-0.55	-0.30
R111	1048.7	1048.7	1049.2	33.0	1049.9	42.0	**1048.7**	1050.1	48.5	0.00	0.00	-0.11	0.09
R112	948.6	948.6	952.2	33.0	948.6	40.0	950.4	954.2	46.7	0.19	0.19	0.19	0.21
C101	827.3	827.3	827.3	29.0	827.3	41.0	**827.3**	827.3	40.4	0.00	0.00	0.00	0.00
C102	827.3	827.3	827.3	32.0	827.3	40.0	**827.3**	827.3	43.1	0.00	0.00	0.00	0.00
C103	826.3	826.3	826.3	34.0	826.3	41.0	**826.3**	826.3	44.9	0.00	0.00	0.00	0.00
C104	822.9	822.9	822.9	36.0	822.9	40.0	**822.9**	822.9	45.9	0.00	0.00	0.00	0.00
C105	827.3	827.3	827.3	30.0	827.3	39.0	**827.3**	827.3	41.3	0.00	0.00	0.00	0.00
C106	827.3	827.3	827.3	31.0	827.3	41.0	**827.3**	827.3	42.5	0.00	0.00	0.00	0.00
C107	827.3	827.3	827.3	31.0	827.3	41.0	**827.3**	827.3	42.6	0.00	0.00	0.00	0.00
C108	827.3	827.3	827.3	32.0	827.3	42.0	**827.3**	827.3	43.5	0.00	0.00	0.00	0.00
C109	827.3	827.3	827.3	34.0	827.3	40.0	**827.3**	827.3	46.0	0.00	0.00	0.00	0.00
RC101	1619.8	1619.8	1629.8	28.0	1619.8	44.0	**1619.8**	1628.8	56.1	0.00	0.00	0.00	-0.06
RC102	1457.4	1463.5	1475.1	30.0	1463.5	42.0	**1457.4**	1470.4	54.4	0.00	-0.42	-0.42	-0.32
RC103	1258.0	1267.0	1272.2	31.0	1267.1	43.0	**1258.0**	1266.9	50.2	0.00	-0.71	-0.72	-0.42
RC104	1132.3	1132.6	1132.8	33.0	1133.1	42.0	1132.6	1134.9	48.3	0.03	0.00	-0.04	0.19
RC105	1513.7	1513.7	1514.2	30.0	1513.7	43.0	**1513.7**	1514.1	56.0	0.00	0.00	0.00	-0.01
RC106	1372.7	1373.9	1376.1	29.0	1372.7	41.0	1373.9	1378.4	50.1	0.09	0.00	0.09	0.17
RC107	1207.8	1209.3	1213.0	30.0	1209.3	40.0	**1207.8**	1212.1	49.1	0.00	-0.12	-0.12	-0.07
RC108	1114.2	1114.2	1124.6	31.0	1114.2	43.0	**1114.2**	1118.2	47.0	0.00	0.00	0.00	-0.57
R201	1143.2	1148.5	1153.9	45.0	1143.2	71.0	1148.1	1152.1	39.6	0.43	-0.03	0.43	-0.16
R202	1029.6	1036.9	1041.0	54.0	1032.2	72.0	1033.8	1038.8	41.9	0.41	-0.30	0.16	-0.21
R203	870.8	872.4	876.5	60.0	873.3	76.0	871.3	874.1	43.7	0.06	-0.13	-0.23	-0.27
R204	731.3	731.3	731.5	67.0	731.3	75.0	731.8	733.5	46.2	0.07	0.07	0.07	0.27
R205	949.8	949.8	952.4	58.0	950.4	71.0	**949.8**	952.8	40.4	0.00	0.00	-0.06	0.04
R206	875.9	880.6	880.6	61.0	881.0	76.0	880.6	882.3	43.0	0.54	0.00	-0.05	0.19
R207	794.0	794.0	796.4	72.0	794.0	85.0	**794.0**	796.0	45.1	0.00	0.00	0.00	-0.05
R208*	701.2	701.2	703.1	86.0	702.9	88.0	**701.0**	705.2	48.6	**-0.03**	-0.03	-0.27	0.30
R209	854.8	855.8	860.2	60.0	854.8	74.0	856.0	861.4	41.6	0.14	0.02	0.14	0.14
R210	900.5	908.4	914.0	59.0	906.3	70.0	908.4	914.7	43.5	0.88	0.00	0.23	0.08
R211	746.7	752.3	758.3	67.0	751.6	74.0	751.7	755.6	45.1	0.67	-0.08	0.01	-0.36
C201	589.1	589.1	589.1	69.0	589.1	82.0	**589.1**	589.1	37.9	0.00	0.00	0.00	0.00
C202	589.1	589.1	589.1	74.0	589.1	85.0	**589.1**	589.1	41.6	0.00	0.00	0.00	0.00
C203	588.7	588.7	588.7	80.0	588.7	92.0	**588.7**	588.7	41.5	0.00	0.00	0.00	0.00
C204	588.1	588.1	588.1	84.0	588.1	91.0	**588.1**	588.1	45.6	0.00	0.00	0.00	0.00
C205	586.4	586.4	586.4	76.0	586.4	86.0	**586.4**	586.4	38.5	0.00	0.00	0.00	0.00
C206	586.0	586.0	586.0	72.0	586.0	81.0	**586.0**	586.0	39.1	0.00	0.00	0.00	0.00
C207	585.8	585.8	585.8	74.0	585.8	86.0	**585.8**	585.8	40.8	0.00	0.00	0.00	0.00
C208	585.8	585.8	585.8	74.0	585.8	88.0	**585.8**	585.8	39.4	0.00	0.00	0.00	0.00
RC201	1261.8	1262.6	1272.3	42.0	1262.7	74.0	1264.4	1271.1	42.1	0.21	0.14	0.13	-0.09
RC202	1092.3	1095.8	1097.4	46.0	1095.8	71.0	1095.6	1098.2	44.5	0.30	-0.02	-0.02	0.07
RC203	923.7	923.7	937.6	56.0	923.7	73.0	926.0	934.5	43.5	0.25	0.25	0.25	-0.33
RC204	783.5	785.8	788.1	68.0	783.8	76.0	**783.5**	785.3	45.2	0.00	-0.29	-0.04	-0.36
RC205	1154.0	1154.0	1154.0	45.0	1154.0	64.0	**1154.0**	1156.1	43.2	0.00	0.00	0.00	0.18
RC206	1051.1	1051.1	1062.5	52.0	1051.1	68.0	1053.3	1065.0	41.7	0.21	0.21	0.21	0.24
RC207	962.9	966.6	976.2	55.0	966.6	64.0	963.3	974.4	43.6	0.04	-0.34	-0.34	-0.18
RC208	776.1	777.3	790.5	65.0	777.3	72.0	777.9	787.3	44.7	0.23	0.08	0.08	-0.40
Average	**973.2**	**974.6**	**977.7**	**47.3**	**974.3**	**59.4**	**974.1**	**977.3**	**45.8**	**0.09**	**-0.04**	**-0.02**	**-0.04**

*** The optimal solution for R208 is not known. We use the best UB reported in the literature.**

Table B.2: Real numbered ALNS results of Solomon's VRPTW benchmark instances.

Real numbered results of Solomon's VRPTW instances

	Best Known	PR			The Proposed ALNS			Deviation from (%)		
Instance	distance	best dist.	avg. dist.	avg. time (secs)	best dist.	avg. dist.	avg. time (secs)	best known	PR best	PR avg.
R101	1642.87	1650.80	1650.86	55.0	1642.88	1644.28	61.2	0.00	-0.48	-0.40
R102	1472.62	1486.12	1486.89	62.0	1473.18	1473.67	59.0	0.04	-0.87	-0.89
R103	1213.62	1292.68	1294.89	64.0	**1213.62**	1213.98	50.7	0.00	-6.12	-6.25
R104	976.61	987.85	1013.13	61.0	981.23	985.15	44.2	0.47	-0.67	-2.76
R105	1360.78	1377.11	1378.77	56.0	**1360.78**	1361.16	51.7	0.00	-1.19	-1.28
R106	1240.26	1252.03	1258.40	61.0	**1239.37**	1239.68	48.8	**-0.07**	-1.01	-1.49
R107	1073.01	1113.70	1118.18	52.0	**1072.12**	1076.46	51.0	**-0.08**	-3.73	-3.73
R108	944.44	963.91	969.37	40.0	**938.20**	947.92	48.2	**-0.66**	-2.67	-2.21
R109	1151.84	1194.73	1213.09	47.0	1151.84	1153.74	48.7	0.00	-3.59	-4.89
R110	1072.41	1119.14	1149.56	41.0	1080.24	1085.59	47.2	0.73	-3.48	-5.56
R111	1053.50	1096.74	1112.14	46.0	1053.50	1054.67	46.2	0.00	-3.94	-5.17
R112	953.63	983.16	1000.60	58.0	958.81	964.58	44.6	0.54	-2.48	-3.60
C101	828.94	828.94	828.94	29.0	**828.94**	828.94	39.2	0.00	0.00	0.00
C102	828.94	828.94	828.94	59.0	**828.94**	828.94	41.7	0.00	0.00	0.00
C103	828.06	828.06	828.06	65.0	**828.06**	828.06	43.2	0.00	0.00	0.00
C104	824.78	824.78	824.78	69.0	**824.78**	824.78	44.3	0.00	0.00	0.00
C105	828.94	828.94	828.94	31.0	**828.94**	828.94	40.0	0.00	0.00	0.00
C106	828.94	828.94	828.94	32.0	**828.94**	828.94	41.1	0.00	0.00	0.00
C107	828.94	828.94	828.94	32.0	**828.94**	828.94	41.0	0.00	0.00	0.00
C108	828.94	828.94	828.94	61.0	**828.94**	828.94	42.4	0.00	0.00	0.00
C109	828.94	828.94	828.94	64.0	**828.94**	828.94	44.5	0.00	0.00	0.00
RC101	1623.58	1688.35	1697.43	53.0	1624.97	1640.43	52.5	0.09	-3.75	-3.36
RC102	1461.23	1547.04	1554.75	56.0	**1461.23**	1469.95	51.1	0.00	-5.55	-5.45
RC103	1261.67	1262.02	1270.78	58.0	**1261.67**	1272.64	46.7	0.00	-0.03	0.15
RC104	1135.48	1135.52	1135.80	60.0	**1135.48**	1136.91	44.1	0.00	0.00	0.10
RC105	1518.58	1629.44	1640.18	54.0	**1518.58**	1519.02	52.4	0.00	-6.80	-7.39
RC106	1376.99	1413.07	1432.12	49.0	**1376.99**	1382.96	48.5	0.00	-2.55	-3.43
RC107	1212.83	1230.95	1232.48	56.0	**1211.11**	1215.55	51.4	**-0.14**	-1.61	-1.37
RC108	1117.53	1140.87	1167.55	41.0	1118.13	1121.95	45.7	0.05	-1.99	-3.91
R201	1147.80	1253.23	1253.23	133.0	1153.37	1159.28	38.2	0.48	-7.97	-7.50
R202	1034.35	1195.30	1229.81	96.0	1037.23	1042.82	40.3	0.28	-13.22	-15.20
R203	874.87	939.58	944.64	164.0	876.25	881.26	42.2	0.16	-6.74	-6.71
R204	735.80	833.09	841.48	182.0	735.86	740.89	44.3	0.01	-11.67	-11.95
R205	954.16	994.43	1018.90	97.0	**954.16**	957.83	39.0	0.00	-4.05	-5.99
R206	879.89	915.27	923.91	192.0	884.85	887.70	41.7	0.56	-3.32	-3.92
R207	797.99	893.33	928.28	180.0	**797.99**	802.00	43.6	0.00	-10.67	-13.60
R208	705.45	726.82	736.12	185.0	707.18	709.82	47.1	0.25	-2.70	-3.57
R209	859.39	914.45	926.72	101.0	861.14	864.92	40.0	0.20	-5.83	-6.67
R210	910.70	954.12	955.02	112.0	**909.96**	917.36	44.0	**-0.08**	-4.63	-3.94
R211	755.82	889.99	925.03	216.0	756.50	760.80	42.6	0.09	-15.00	-17.75
C201	591.56	591.56	591.56	78.0	**591.56**	591.56	37.1	0.00	0.00	0.00
C202	591.56	591.56	591.56	88.0	**591.56**	591.56	40.6	0.00	0.00	0.00
C203	591.17	591.17	591.17	96.0	**591.17**	591.17	40.4	0.00	0.00	0.00
C204	590.60	590.60	590.60	102.0	**590.60**	590.60	44.3	0.00	0.00	0.00
C205	588.88	588.88	588.88	81.0	**588.88**	588.88	37.4	0.00	0.00	0.00
C206	588.49	588.49	588.49	83.0	**588.49**	588.49	38.1	0.00	0.00	0.00
C207	588.29	588.29	588.29	84.0	**588.29**	588.29	39.8	0.00	0.00	0.00
C208	588.32	588.32	588.32	85.0	**588.32**	588.32	38.2	0.00	0.00	0.00
RC201	1265.56	1413.52	1417.80	83.0	1267.83	1277.67	39.9	0.18	-10.31	-9.88
RC202	1095.64	1368.04	1405.16	96.0	1098.81	1103.34	41.0	0.29	-19.68	-21.48
RC203	926.89	1068.08	1075.51	100.0	**926.91**	937.12	41.4	0.00	-13.22	-12.87
RC204	786.38	799.27	818.00	228.0	786.54	788.68	42.9	0.02	-1.59	-3.58
RC205	1157.55	1302.42	1318.01	134.0	**1157.55**	1159.37	40.2	0.00	-11.12	-12.04
RC206	1054.61	1146.32	1155.91	87.0	1058.06	1069.90	38.4	0.33	-7.70	-7.44
RC207	966.08	1070.85	1095.29	96.0	974.48	979.02	40.1	0.87	-9.00	-10.62
RC208	779.31	829.69	834.83	109.0	780.72	790.20	41.4	0.18	-5.90	-5.35
Average	**977.25**	**1022.27**	**1031.34**	**85.7**	**978.10**	**981.51**	**44.2**	**0.09**	**-3.87**	**-4.34**

Appendix C: Routes of the improved instances

We report here the new best solutions we obtained for truncated numbered problem R208, and real numbered problems R106, R107, R108, R210 and RC107.

Table C.1: Improved routes for Solomon's VRPTW instances with 100 customers and tight time windows

Route		Cost
Real number R106. Total Cost = 1239.37		
1	0-94-92-42-15-57-87-97-95-13-0	61.64
2	0-12-29-78-79-68-54-24-80-0	86.18
3	0-69-30-51-81-9-35-34-3-77-0	106.17
4	0-73-41-22-75-56-74-2-58-0	127.24
5	0-48-47-36-19-49-46-82-7-52-0	127.23
6	0-27-62-88-18-89-0	75.88
7	0-21-72-39-23-67-55-4-25-26-0	79.70
8	0-63-64-11-90-10-31-0	126.94
9	0-59-37-14-44-38-86-43-100-98-93-0	129.21
10	0-28-76-40-53-0	46.17
11	0-96-85-91-16-61-99-6-0	62.65
12	0-83-45-8-84-17-5-60-0	106.13
13	0-50-33-65-71-66-20-32-70-1-0	104.24
Real number R107. Total Cost = 1072.12		
1	0-60-83-45-46-8-84-5-17-61-85-93-0	113.47
2	0-94-96-92-59-99-6-87-13-0	62.75
3	0-27-69-30-88-31-10-70-1-0	86.16
4	0-33-81-65-71-9-35-34-3-77-0	126.47
5	0-52-7-62-11-63-90-32-66-20-51-50-0	114.94
6	0-2-57-43-15-41-22-75-56-74-72-73-21-0	103.08
7	0-95-97-42-14-44-38-86-16-91-100-37-98-0	105.74
8	0-48-47-36-64-49-19-82-18-89-0	126.00
9	0-53-40-58-0	24.36
10	0-26-39-23-67-55-4-25-54-0	127.04
11	0-28-76-79-78-29-24-68-80-12-0	82.10
Real number R108. Total Cost = 938.20		
1	0-2-57-15-43-42-87-97-95-94-13-58-0	124.44
2	0-73-22-41-23-67-39-56-75-74-72-21-40-0	107.11
3	0-6-96-59-99-93-5-84-17-45-83-60-89-0	107.75
4	0-52-88-62-19-11-64-63-90-32-10-31-0	90.26
5	0-26-12-80-68-29-24-55-4-25-54-0	78.99
6	0-27-69-50-76-3-79-78-34-81-33-77-28-0	115.17
7	0-1-70-30-51-9-35-71-65-66-20-0	84.67
8	0-92-98-91-44-14-38-86-16-61-85-100-37-0	114.80
9	0-18-7-82-8-46-36-49-47-48-0	106.08
10	0-53-0	8.94
Real number RC107. Total Cost = 1211.11		
1	0-72-71-93-94-67-50-62-91-80-0	105.63
2	0-41-38-39-42-44-43-40-37-35-36-0	108.66
3	0-11-12-14-47-17-16-15-13-9-10-0	100.98
4	0-92-95-84-85-63-51-76-89-56-0	123.47
5	0-69-98-88-53-78-73-79-60-55-70-68-0	116.69
6	0-2-6-7-8-5-3-1-45-46-4-100-0	101.80
7	0-31-29-27-28-26-30-32-34-33-0	140.98
8	0-82-99-52-87-59-86-57-66-0	96.79
9	0-65-83-25-77-75-97-58-74-0	150.63
10	0-61-81-54-96-0	51.72
11	0-64-22-19-23-21-18-48-49-20-24-0	105.27
12	0-90-0	8.49

Table C.2: Improved routes for Solomon's VRPTW instances with 100 customers and wide time windows

Route		Cost
Real number R210. Total Cost = 909.96		
1	0-95-92-59-5-83-45-36-47-48-82-18-7-88-62-19-11-63-64-49-46-8-84-17-85-98-37-100-91-93-60-89-0	274.50
2	0-21-73-72-23-67-39-56-75-22-41-74-4-55-25-54-26-0	145.37
3	0-27-69-1-30-51-33-71-65-66-20-32-90-10-70-31-52-0	152.31
4	0-28-12-76-3-79-29-78-81-9-35-34-24-80-68-77-50-0	144.95
5	0-2-57-15-42-14-44-38-86-16-61-99-96-6-94-87-43-97-13-0	168.47
6	0-53-40-58-0	24.36
Truncated number R208. Total cost = 701		
1	0-52-18-82-48-7-88-31-70-30-32-90-63-10-62-19-11-64-49-36-47-46-8-45-17-84-83-60-5-99-96-97-87-37-98-91-100-42-57-2-13-0	290.6
2	0-27-69-1-50-76-33-81-9-51-20-66-65-71-35-34-78-79-3-77-68-80-29-24-54-12-26-28-0	192.4
3	0-89-6-94-95-92-59-93-85-61-16-86-44-38-14-43-15-41-22-75-56-23-67-39-25-55-4-72-74-73-21-40-58-0	209.2
4	0-53-0	8.8

Appendix D: Detailed ALNS results of new benchmark instances with 25 nodes

We report here the detailed ALNS results of new benchmark instances with 25 nodes. In the following tables; $gap = 100 * (upper_{bound} - lower_{bound})/lower_{bound}$, $best_{gap} = 100 * (best_{dist} - upper_{bound})/upper_{bound}$, $avg._{gap} = 100 * (avg._{dist} - upper_{bound})/upper_{bound}$.

Table D.1: ALNS results for 25-node problems using Category 1 parameters

	CPLEX					The Proposed ALNS					
Instances	lower bound	upper bound	gap (%)	#veh.	sol. time (secs)	best dist.	best gap (%)	best #veh.	avg. dist.	avg. gap (%)	avg. time (secs)
M-R101	615.3	711.6	15.65	9	10800	711.6	0.00	9	711.6	0.00	7.9
M-R102	370.3	567.1	53.15	7	10800	555.9	-1.97	7	555.9	-1.97	9.1
M-R103	-	-	-	-	-	484.0	-	5	484.0	-	7
M-R104	304.7	433.6	42.30	4	10800	433.6	0.00	4	433.6	0.00	7.4
M-R105	455.1	585.4	28.63	6	2316	585.4	0.00	6	585.4	0.00	10.5
M-R106	338.9	575.2	69.73	5	10800	556.5	-3.25	6	556.5	-3.25	7
M-R107	308.0	459.4	49.16	5	10800	440.5	-4.11	4	440.5	-4.11	8
M-R108	-	-	-	-	-	420.7	-	3	420.7	-	7.4
M-R109	357.9	519.1	45.04	5	10800	490.1	-5.59	5	490.1	-5.59	8.4
M-R110	-	-	-	-	-	541.3	-	5	541.3	-	8
M-R111	348.9	497.4	42.56	5	2274	451.3	-9.27	4	451.3	-9.27	7.6
M-R112	-	-	-	-	-	409.6	-	4	409.6	-	6.9
M-C101	239.5	239.5	0.00	3	896	239.5	0.00	3	239.5	0.00	7.2
M-C102	171.9	195.3	13.64	3	10800	195.3	0.00	3	195.3	0.00	7.6
M-C103	199.9	199.9	0.00	3	2723	199.9	0.00	3	199.9	0.00	8
M-C104	148.4	246.3	65.97	3	10800	233.1	-5.36	3	233.1	-5.36	9.1
M-C105	268.7	268.7	0.00	4	95	268.7	0.00	4	268.7	0.00	7.3
M-C106	234.4	234.4	0.00	4	136	234.4	0.00	4	234.4	0.00	8
M-C107	269.6	269.6	0.00	4	6023	269.6	0.00	4	269.6	0.00	8.4
M-C108	193.2	193.2	0.00	3	6897	193.2	0.00	3	193.2	0.00	9.1
M-C109	142.8	209.7	46.85	3	10800	209.7	0.00	3	209.7	0.00	8.1
M-RC101	334.7	456.2	36.30	4	10800	456.2	0.00	4	456.2	0.00	5.8
M-RC102	326.4	574.7	76.07	5	10800	574.7	0.00	5	574.7	0.00	6.8
M-RC103	-	-	-	-	-	537.0	-	4	537.0	-	6
M-RC104	-	-	-	-	-	440.0	-	4	440.0	-	5.9
M-RC105	308.2	546.0	77.16	5	10800	528.2	-3.26	5	528.2	-3.26	7.1
M-RC106	321.2	489.5	52.40	4	10800	489.5	0.00	4	489.5	0.00	6.4
M-RC107	-	-	-	-	-	417.5	-	4	417.5	-	6.7
M-RC108	-	-	-	-	-	461.0	-	4	461.0	-	6.3
M-R201	466.4	476.2	2.10	3	10800	476.2	0.00	3	476.2	0.00	7
M-R202	329.7	455.6	38.19	3	10800	410.9	-9.81	2	410.9	-9.81	7
M-R203	333.9	394.6	18.18	3	10800	384.2	-2.64	2	384.2	-2.64	6.7
M-R204	308.1	387.5	25.77	2	10800	387.2	-0.08	2	387.2	-0.08	6.2
M-R205	374.2	398.5	6.49	3	10800	398.5	0.00	3	398.5	0.00	6.6
M-R206	316.4	389.6	23.14	2	10800	377.8	-3.03	2	377.8	-3.03	7
M-R207	304.4	357.8	17.54	3	8845	350.6	-2.01	2	350.6	-2.01	6.5
M-R208	298.8	350.3	17.24	2	5359	326.1	-6.91	1	326.8	-6.71	7.1
M-R209	315.7	425.3	34.72	3	6962	388.0	-8.77	2	388.0	-8.77	6.1
M-R210	315.5	573.4	81.74	3	2162	421.5	-26.49	2	421.5	-26.49	7.2
M-R211	317.4	353.7	11.44	2	4875	353.7	0.00	2	353.7	0.00	6.5
M-C201	212.3	212.3	0.00	2	2	212.3	0.00	2	212.3	0.00	7.6
M-C202	280.5	280.5	0.00	2	1044	280.5	0.00	2	280.5	0.00	7.4
M-C203	181.6	245.2	35.02	2	10800	240.7	-1.84	1	240.7	-1.84	8.2
M-C204	182.5	228.5	25.21	2	10800	227.2	-0.57	1	227.4	-0.48	8.6
M-C205	244.7	244.7	0.00	2	2594	244.7	0.00	2	244.7	0.00	8
M-C206	228.4	228.4	0.00	2	3298	228.4	0.00	2	228.4	0.00	7.2
M-C207	259.0	265.0	2.32	2	10800	265.0	0.00	2	265.0	0.00	9.3
M-C208	203.6	226.1	11.05	2	10800	226.1	0.00	2	226.1	0.00	8.7
M-RC201	300.5	450.8	50.02	3	5419	450.8	0.00	3	450.8	0.00	7.4
M-RC202	193.3	396.4	105.07	3	2674	371.5	-6.28	2	371.5	-6.28	5.6
M-RC203	163.9	361.0	120.26	2	1996	353.6	-2.05	2	353.6	-2.05	6.2
M-RC204	161.6	387.9	140.04	2	1694	321.0	-17.25	2	321.2	-17.20	6
M-RC205	204.1	591.2	189.66	3	1919	406.4	-31.26	3	406.4	-31.26	6.5
M-RC206	210.7	370.5	75.84	2	4633	359.7	-2.91	2	359.7	-2.91	5.8
M-RC207	161.8	565.4	249.44	2	1705	349.2	-38.24	2	349.2	-38.24	6.1
M-RC208	154.4	263.4	70.60	2	3133	263.4	0.00	2	263.4	0.00	6.4
Average	276.7	382.3	43.03	3.3	6834.9	376.8	-4.02	3.2	376.9	-4.01	7.3

Table D.2: ALNS results for 25-node problems using Category 2 parameters

Instances	CPLEX lower bound	upper bound	gap (%)	#veh.	sol. time (secs)	The Proposed ALNS best dist.	best gap (%)	best #veh.	avg. dist.	avg. gap (%)	avg. time (secs)
M-R101	590.7	711.6	20.47	9	10800	711.6	0.00	9	711.6	0.00	4.8
M-R102	371.8	555.9	49.52	7	10800	555.9	0.00	7	555.9	0.00	5.7
M-R103	-	-	-	-	-	497.0	-	5	497.0	-	4.6
M-R104	319.6	433.6	35.67	4	10800	433.6	0.00	4	433.6	0.00	4.3
M-R105	-	-	-	-	-	806.5	-	8	806.5	-	6.5
M-R106	-	-	-	-	-	656.8	-	6	656.8	-	5
M-R107	308.2	452.6	46.85	4	10800	452.6	0.00	4	452.6	0.00	4.9
M-R108	-	-	-	-	-	420.7	-	3	420.7	-	4
M-R109	341.5	490.1	43.51	5	10800	490.1	0.00	5	490.1	0.00	4.4
M-R110	-	-	-	-	-	551.2	-	5	551.2	-	4.7
M-R111	331.8	451.3	36.02	4	10800	451.3	0.00	4	451.3	0.00	4.4
M-R112	285.1	456.6	60.15	4	10800	437.7	-4.14	4	437.7	-4.14	4.6
M-C101	191.5	238.7	24.65	3	10800	238.7	0.00	3	238.7	0.00	5
M-C102	153.7	223.5	45.41	3	10800	223.5	0.00	3	223.5	0.00	4.9
M-C103	103.1	199.9	93.89	3	10800	199.9	0.00	3	199.9	0.00	4.8
M-C104	139.0	251.0	80.58	3	10800	232.1	-7.53	3	232.1	-7.53	5.5
M-C105	226.5	275.2	21.50	4	10800	275.2	0.00	4	275.2	0.00	5.4
M-C106	213.8	293.3	37.18	4	10800	293.3	0.00	4	293.3	0.00	4.9
M-C107	159.0	269.6	69.56	4	10800	269.6	0.00	4	269.6	0.00	4.3
M-C108	118.9	327.5	175.44	4	10800	303.9	-7.21	4	303.9	-7.21	5.4
M-C109	108.9	255.9	134.99	3	10800	251.4	-1.76	3	251.5	-1.72	4.8
M-RC101	317.6	528.8	66.50	5	10800	528.8	0.00	5	528.8	0.00	4.5
M-RC102	250.5	582.4	132.50	5	10800	582.4	0.00	5	582.4	0.00	5.2
M-RC103	-	-	-	-	-	537.0	-	4	537.2	-	4.4
M-RC104	-	-	-	-	-	586.2	-	4	586.2	-	4.8
M-RC105	-	-	-	-	-	540.1	-	5	540.1	-	5.5
M-RC106	244.1	489.5	100.53	4	10800	489.5	0.00	4	489.5	0.00	4.9
M-RC107	-	-	-	-	-	539.2	-	4	539.2	-	5
M-RC108	-	-	-	-	-	510.2	-	4	510.2	-	5.1
M-R201	392.1	476.2	21.45	3	10800	476.2	0.00	3	476.2	0.00	4.7
M-R202	300.9	454.5	51.05	3	10800	426.2	-6.23	3	426.7	-6.12	4.5
M-R203	303.0	428.1	41.29	2	10800	384.2	-10.25	2	384.2	-10.25	4.3
M-R204	289.2	394.6	36.45	2	10800	387.2	-1.88	2	387.2	-1.88	4.5
M-R205	338.4	411.2	21.51	2	10800	411.2	0.00	2	411.2	0.00	4.9
M-R206	289.6	425.1	46.79	2	10800	398.2	-6.33	1	398.5	-6.26	5.1
M-R207	291.8	387.8	32.90	1	10800	367.1	-5.34	2	367.1	-5.34	4.8
M-R208	289.4	349.5	20.77	2	10800	344.5	-1.43	1	344.5	-1.43	5.6
M-R209	316.5	404.6	27.84	3	10800	404.1	-0.12	2	404.1	-0.12	4.6
M-R210	320.1	481.5	50.42	2	10800	463.9	-3.66	2	464.7	-3.49	5.8
M-R211	291.5	353.7	21.34	2	10800	353.7	0.00	2	353.7	0.00	5.6
M-C201	212.3	212.3	0.00	2	13	212.3	0.00	2	212.3	0.00	4.8
M-C202	184.5	282.1	52.90	2	10800	282.1	0.00	2	282.1	0.00	6
M-C203	170.1	247.6	45.56	2	10800	240.7	-2.79	1	241.1	-2.63	5
M-C204	163.8	228.5	39.50	2	10800	228.5	0.00	2	228.5	0.00	5.7
M-C205	191.7	244.7	27.65	2	10800	244.7	0.00	2	244.7	0.00	5.3
M-C206	207.5	261.4	25.98	2	10800	261.4	0.00	2	261.4	0.00	5.8
M-C207	223.0	265.0	18.83	2	10800	265.0	0.00	2	265.0	0.00	4.9
M-C208	179.6	227.3	26.56	1	10800	227.3	0.00	1	227.3	0.00	4.9
M-RC201	278.9	457.1	63.89	3	10800	457.1	0.00	3	457.1	0.00	6.1
M-RC202	182.2	383.9	110.70	2	10800	380.1	-0.99	2	380.1	-0.99	4.8
M-RC203	166.1	373.6	124.92	2	10800	353.6	-5.35	2	353.6	-5.35	4.9
M-RC204	158.5	321.0	102.52	2	10800	321.0	0.00	2	321.0	0.00	5.1
M-RC205	192.5	515.7	167.90	3	10800	431.3	-16.37	3	431.3	-16.37	4.9
M-RC206	193.1	359.7	86.28	2	10800	359.7	0.00	2	359.7	0.00	4.7
M-RC207	153.4	387.1	152.35	3	10800	349.2	-9.79	2	349.2	-9.79	4.7
M-RC208	145.6	263.4	80.91	2	10800	263.4	0.00	2	263.4	0.00	5.3
Average	243.5	371.4	60.29	3.0	10566	399.3	-1.98	3.3	399.3	-1.97	5.0

Table D.3: ALNS results for 25-node problems using Category 3 parameters

	CPLEX					The Proposed ALNS					
Instances	lower bound	upper bound	gap (%)	#veh.	sol. time (secs)	best dist.	best gap (%)	best #veh.	avg. dist.	avg. gap (%)	avg. time (secs)
M-R101	590.5	681.2	15.36	8	10800	681.2	0.00	8	681.2	0.00	4.7
M-R102	391.1	606.6	55.10	7	10800	596.1	-1.73	7	596.1	-1.73	5.6
M-R103	-	-	-	-	-	593.4	-	6	593.4	-	5
M-R104	-	-	-	-	-	475.4	-	4	475.4	-	5.3
M-R105	448.9	666.2	48.41	7	10800	642.8	-3.51	7	642.8	-3.51	6.2
M-R106	-	-	-	-	-	709.8	-	6	709.8	-	6.2
M-R107	315.8	459.0	45.35	4	10800	459.0	0.00	4	459.0	0.00	5
M-R108	296.6	416.4	40.39	4	10800	416.4	0.00	4	416.4	0.00	4.3
M-R109	354.5	546.6	54.19	5	10800	546.6	0.00	5	546.6	0.00	5
M-R110	328.9	551.2	67.59	5	10800	551.2	0.00	5	551.2	0.00	4.7
M-R111	331.6	451.3	36.10	4	10800	451.3	0.00	4	451.3	0.00	4.5
M-R112	-	-	-	-	-	437.7	-	4	437.7	-	4.5
M-C101	190.4	239.5	25.79	3	10800	239.5	0.00	3	239.5	0.00	5
M-C102	156.4	240.2	53.58	3	10800	240.2	0.00	3	240.2	0.00	5
M-C103	103.7	199.9	92.77	3	10800	199.9	0.00	3	199.9	0.00	4.7
M-C104	139.3	248.4	78.32	3	10800	227.3	-8.49	3	227.3	-8.49	5.3
M-C105	243.5	243.5	0.00	3	836	243.5	0.00	3	243.5	0.00	4.7
M-C106	233.4	351.2	50.47	5	10800	351.2	0.00	5	351.2	0.00	5.2
M-C107	161.9	269.6	66.52	4	10800	269.6	0.00	4	269.6	0.00	4.4
M-C108	-	-	-	-	-	312.4	-	4	312.4	-	6.3
M-C109	115.2	245.4	113.02	3	10800	216.6	-11.74	3	216.6	-11.74	4.4
M-RC101	288.8	726.6	151.59	6	10800	726.6	0.00	6	726.7	0.01	5.6
M-RC102	301.5	584.9	94.00	5	10800	584.9	0.00	5	584.9	0.00	6.1
M-RC103	-	-	-	-	-	587.8	-	4	587.8	-	5.1
M-RC104	-	-	-	-	-	710.9	-	5	710.9	-	5.9
M-RC105	239.2	543.8	127.34	5	10800	543.8	0.00	5	543.8	0.00	5.4
M-RC106	243.7	440.4	80.71	4	10800	440.4	0.00	4	440.4	0.00	4.8
M-RC107	-	-	-	-	-	503.8	-	4	503.8	-	5.1
M-RC108	-	-	-	-	-	532.0	-	4	532.0	-	4.9
M-R201	394.9	474.8	20.23	3	10800	474.8	0.00	3	474.8	0.00	5.4
M-R202	297.8	391.6	31.50	3	10800	391.6	0.00	3	391.6	0.00	4.5
M-R203	307.1	390.3	27.09	3	10800	384.2	-1.56	2	384.2	-1.56	4.2
M-R204	293.8	387.2	31.79	2	10800	387.2	0.00	2	387.2	0.00	4.4
M-R205	340.3	411.2	20.83	2	10800	411.2	0.00	2	411.2	0.00	4.9
M-R206	286.6	377.8	31.82	2	10800	377.8	0.00	2	377.8	0.00	5.2
M-R207	296.2	403.1	36.09	2	10800	367.1	-8.93	2	369.5	-8.34	5.3
M-R208	291.9	329.4	12.85	2	10800	326.5	-0.88	1	326.5	-0.88	5.2
M-R209	314.6	453.9	44.28	3	10800	446.9	-1.54	2	447.2	-1.48	4.1
M-R210	315.8	463.5	46.77	2	10800	452.6	-2.35	3	454.0	-2.05	6.2
M-R211	293.4	376.2	28.22	2	10800	362.4	-3.67	1	362.4	-3.67	6.2
M-C201	212.3	212.3	0.00	2	22	212.3	0.00	2	212.3	0.00	4.9
M-C202	233.1	256.9	10.21	2	10800	256.9	0.00	2	256.9	0.00	5.5
M-C203	172.0	240.3	39.71	2	10800	240.3	0.00	2	240.3	0.00	6
M-C204	162.5	228.5	40.62	2	10800	227.2	-0.57	1	227.6	-0.39	5.6
M-C205	221.5	294.7	33.05	2	10800	294.7	0.00	2	294.7	0.00	5.5
M-C206	205.1	261.4	27.45	2	10800	261.4	0.00	2	261.4	0.00	5.8
M-C207	231.2	249.4	7.87	2	10800	249.4	0.00	2	249.4	0.00	5.3
M-C208	181.8	228.2	25.52	1	10800	228.2	0.00	1	228.2	0.00	5.8
M-RC201	295.6	457.1	54.63	3	10800	457.1	0.00	3	457.1	0.00	5.6
M-RC202	187.8	434.1	131.15	2	10800	410.6	-5.41	2	410.6	-5.41	5
M-RC203	-	-	-	-	-	353.6	-	2	353.6	-	4.9
M-RC204	-	-	-	-	-	321.0	-	2	321.0	-	5
M-RC205	197.5	431.3	118.38	3	10800	431.3	0.00	3	431.3	0.00	4.8
M-RC206	195.8	432.4	120.84	2	10800	359.7	-16.81	2	359.7	-16.81	4.5
M-RC207	158.5	394.9	149.15	3	10800	365.9	-7.34	2	365.9	-7.34	4.8
M-RC208	147.0	263.4	79.18	2	10800	263.4	0.00	2	263.4	0.00	5.8
Average	260.2	390.1	54.80	3.3	10339	407.3	-1.66	3.3	407.3	-1.63	5.2

Table D.4: ALNS results for 25-node problems using Category 4 parameters

	CPLEX					The Proposed ALNS					
Instances	**lower bound**	**upper bound**	**gap (%)**	**#veh.**	**sol. time (secs)**	**best dist.**	**best gap (%)**	**best #veh.**	**avg. dist.**	**avg. gap (%)**	**avg. time (secs)**
M-R101	437.8	646.2	47.60	8	10800	646.2	0.00	8	646.2	0.00	5.9
M-R102	-	-	-	-	-	546.0	-	7	546.0	-	6.1
M-R103	-	-	-	-	-	494.3	-	5	494.3	-	5.6
M-R104	-	-	-	-	-	429.8	-	4	429.8	-	5.7
M-R105	367.3	507.3	38.12	6	10800	507.3	0.00	6	507.3	0.00	5.7
M-R106	315.5	504.2	59.81	5	10800	477.4	-5.32	4	477.4	-5.32	5.4
M-R107	275.9	409.3	48.35	4	10800	403.9	-1.32	4	403.9	-1.32	5.1
M-R108	-	-	-	-	-	444.6	-	4	444.6	-	5.6
M-R109	311.7	530.0	70.04	6	10800	502.3	-5.23	5	502.3	-5.23	5.2
M-R110	-	-	-	-	-	574.6	-	5	574.6	-	5.7
M-R111	298.7	457.3	53.10	5	10800	453.4	-0.85	5	453.4	-0.85	5.4
M-R112	-	-	-	-	-	483.2	-	4	483.2	-	5.9
M-C101	176.2	262.0	48.69	4	10800	262.0	0.00	4	262.0	0.00	5.8
M-C102	-	-	-	-	-	245.2	-	3	245.2	-	5.3
M-C103	102.5	188.5	83.90	3	10800	188.5	0.00	3	188.5	0.00	6
M-C104	130.6	189.5	45.10	3	10800	189.3	-0.11	3	189.3	-0.11	6
M-C105	180.2	262.8	45.84	4	10800	262.8	0.00	4	262.8	0.00	6.7
M-C106	212.3	281.9	32.78	5	10800	281.9	0.00	5	282.3	0.14	5.5
M-C107	141.4	265.0	87.41	3	10800	256.3	-3.28	3	256.8	-3.09	5.8
M-C108	123.8	216.9	75.20	3	10800	216.9	0.00	3	216.9	0.00	5.9
M-C109	102.0	266.2	160.98	4	10800	235.7	-11.46	3	235.7	-11.46	5.6
M-RC101	242.9	536.3	120.79	5	10800	536.3	0.00	5	536.3	0.00	5.4
M-RC102	-	-	-	-	-	584.9	-	5	584.9	-	5.8
M-RC103	-	-	-	-	-	397.9	-	3	397.9	-	5.7
M-RC104	-	-	-	-	-	310.6	-	3	310.6	-	6
M-RC105	-	-	-	-	-	586.9	-	5	586.9	-	5.8
M-RC106	-	-	-	-	-	546.7	-	5	546.7	-	6.6
M-RC107	-	-	-	-	-	299.4	-	3	299.4	-	5.1
M-RC108	-	-	-	-	-	447.3	-	4	455.2	-	6.8
M-R201	348.5	415.3	19.17	3	10800	415.3	0.00	3	415.3	0.00	5.8
M-R202	275.2	361.0	31.18	3	10800	359.1	-0.53	2	359.1	-0.53	5.4
M-R203	289.5	367.9	27.08	2	10800	364.9	-0.82	2	364.9	-0.82	6.1
M-R204	276.5	351.2	27.02	3	10800	350.7	-0.14	2	350.7	-0.14	5.4
M-R205	305.8	401.9	31.43	3	10800	386.0	-3.96	3	386.0	-3.96	6.3
M-R206	269.6	362.8	34.57	1	10800	359.1	-1.02	2	361.5	-0.36	6.3
M-R207	272.4	345.5	26.84	3	10800	340.7	-1.39	2	340.7	-1.39	5.9
M-R208	262.9	308.8	17.46	1	10800	303.6	-1.68	1	305.4	-1.10	6.2
M-R209	298.5	388.2	30.05	2	10800	374.7	-3.48	2	374.7	-3.48	5.8
M-R210	282.5	391.1	38.44	2	10800	388.0	-0.79	2	388.0	-0.79	5.9
M-R211	262.6	397.4	51.33	3	10800	341.9	-13.97	2	341.9	-13.97	5.9
M-C201	205.5	273.3	32.99	2	10800	273.3	0.00	2	273.3	0.00	6.4
M-C202	179.0	213.5	19.27	2	10800	213.5	0.00	2	213.5	0.00	6.3
M-C203	152.9	260.8	70.57	1	10800	225.2	-13.65	2	225.2	-13.65	7
M-C204	152.0	200.6	31.97	1	10800	198.4	-1.10	1	198.4	-1.10	5.9
M-C205	185.5	264.7	42.70	2	10800	264.7	0.00	2	264.7	0.00	6.3
M-C206	169.7	241.5	42.31	2	10800	238.0	-1.45	2	238.0	-1.45	5.9
M-C207	158.1	213.4	34.98	2	10800	213.4	0.00	2	213.4	0.00	6
M-C208	164.6	252.4	53.34	1	10800	252.4	0.00	1	252.4	0.00	5.4
M-RC201	182.4	424.4	132.68	3	10800	424.4	0.00	3	424.4	0.00	5.4
M-RC202	157.6	394.1	150.06	2	10800	394.1	0.00	2	394.1	0.00	6.8
M-RC203	157.6	380.2	141.24	2	10800	342.2	-9.99	2	342.2	-9.99	5.6
M-RC204	153.2	306.8	100.26	1	10800	306.8	0.00	1	306.8	0.00	7
M-RC205	172.5	449.8	160.75	2	10800	359.1	-20.16	3	359.1	-20.16	6.4
M-RC206	164.3	361.2	119.84	3	10800	354.7	-1.80	2	354.7	-1.80	5.6
M-RC207	148.6	412.5	177.59	2	10800	298.5	-27.64	2	298.5	-27.64	5.8
M-RC208	142.4	288.6	102.67	2	10800	288.6	0.00	2	288.6	0.00	7.4
Average	219.3	346.5	65.85	3.0	10800	365.1	-3.12	3.2	365.3	-3.08	5.9

Table D.5: ALNS results for 25-node problems using Category 5 parameters

Instances	CPLEX					The Proposed ALNS					
	lower bound	upper bound	gap (%)	#veh.	sol. time (secs)	best dist.	best gap (%)	best #veh.	avg. dist.	avg. gap (%)	avg. time (secs)
M-R101	434.7	635.1	46.10	8	10800	635.1	0.00	8	635.1	0.00	6
M-R102	340.0	586.7	72.56	7	10800	578.2	-1.45	7	578.2	-1.45	5.9
M-R103	-	-	-	-	-	494.9	-	5	494.9	-	5.6
M-R104	303.8	461.3	51.84	5	10800	461.3	0.00	5	461.3	0.00	6.9
M-R105	391.3	512.4	30.95	6	10800	512.4	0.00	6	512.4	0.00	5.9
M-R106	-	-	-	-	-	477.4	-	4	477.4	-	5.8
M-R107	294.8	430.3	45.96	4	10800	430.3	0.00	4	430.3	0.00	5.5
M-R108	-	-	-	-	-	396.4	-	3	396.4	-	5.6
M-R109	-	-	-	-	-	545.7	-	5	545.7	-	5.8
M-R110	-	-	-	-	-	623.6	-	6	623.6	-	5.7
M-R111	-	-	-	-	-	462.5	-	5	462.5	-	5.6
M-R112	-	-	-	-	-	634.9	-	5	634.9	-	6.7
M-C101	183.5	303.3	65.29	4	10800	303.3	0.00	4	303.3	0.00	6.9
M-C102	150.7	277.0	83.81	4	10800	267.0	-3.61	3	267.0	-3.61	5.7
M-C103	101.5	188.5	85.71	3	10800	188.5	0.00	3	188.5	0.00	6.4
M-C104	132.0	195.2	47.88	3	10800	195.2	0.00	3	195.2	0.00	6.3
M-C105	153.2	324.2	111.62	4	10800	323.7	-0.15	4	323.7	-0.15	6.8
M-C106	215.8	281.9	30.63	5	10800	281.9	0.00	5	281.9	0.00	6
M-C107	146.2	265.0	81.26	3	10800	265.0	0.00	3	265.5	0.19	5.9
M-C108	114.9	249.8	117.41	3	10800	249.8	0.00	3	249.8	0.00	6.7
M-C109	-	-	-	-	-	317.7	-	3	317.8	-	6.9
M-RC101	229.1	536.3	134.09	5	10800	536.3	0.00	5	536.3	0.00	5.5
M-RC102	-	-	-	-	-	584.9	-	5	584.9	-	6.2
M-RC103	-	-	-	-	-	390.4	-	3	390.4	-	6
M-RC104	-	-	-	-	-	490.7	-	4	490.7	-	7.1
M-RC105	-	-	-	-	-	495.0	-	5	495.0	-	5.7
M-RC106	-	-	-	-	-	622.0	-	5	622.0	-	7.3
M-RC107	-	-	-	-	-	400.9	-	3	400.9	-	5.7
M-RC108	-	-	-	-	-	372.2	-	3	372.2	-	6.1
M-R201	356.0	421.1	18.29	3	10800	421.1	0.00	3	421.1	0.00	5.8
M-R202	275.5	413.5	50.09	3	10800	394.5	-4.59	2	394.5	-4.59	6
M-R203	294.0	372.8	26.80	2	10800	372.8	0.00	2	372.8	0.00	6.1
M-R204	275.3	357.2	29.75	3	10800	354.0	-0.90	2	354.0	-0.90	5.4
M-R205	312.3	390.8	25.14	3	10800	390.8	0.00	3	390.8	0.00	7.4
M-R206	276.7	369.6	33.57	1	10800	369.6	0.00	1	369.6	0.00	6.6
M-R207	274.3	360.3	31.35	2	10800	349.6	-2.97	2	350.0	-2.86	6.2
M-R208	269.0	303.6	12.86	1	10800	303.6	0.00	1	304.4	0.26	6.4
M-R209	304.1	393.2	29.30	2	10800	393.2	0.00	2	393.4	0.05	6.1
M-R210	294.7	391.6	32.88	2	10800	388.5	-0.79	2	388.5	-0.79	6.8
M-R211	272.3	368.5	35.33	2	10800	341.9	-7.22	2	341.9	-7.22	6.3
M-C201	213.3	273.3	28.13	2	10800	273.3	0.00	2	273.3	0.00	6.5
M-C202	176.9	213.5	20.69	2	10800	213.5	0.00	2	213.5	0.00	6.7
M-C203	163.6	227.2	38.88	1	10800	227.2	0.00	1	227.2	0.00	7.8
M-C204	154.5	202.1	30.81	1	10800	202.1	0.00	1	202.1	0.00	6
M-C205	200.5	264.7	32.02	2	10800	264.7	0.00	2	264.7	0.00	6.6
M-C206	172.0	258.4	50.23	2	10800	258.4	0.00	2	258.4	0.00	5.8
M-C207	157.4	215.6	36.98	2	10800	215.6	0.00	2	215.6	0.00	6.2
M-C208	161.4	215.7	33.64	2	10800	215.7	0.00	2	215.7	0.00	5.7
M-RC201	188.4	408.7	116.93	3	10800	368.8	-9.76	3	368.8	-9.76	5.9
M-RC202	158.7	394.1	148.33	2	10800	394.1	0.00	2	394.1	0.00	6.9
M-RC203	157.9	342.4	116.85	2	10800	342.4	0.00	2	342.4	0.00	7.5
M-RC204	153.9	306.8	99.35	1	10800	306.8	0.00	1	306.8	0.00	7.3
M-RC205	180.1	498.8	176.96	2	10800	422.6	-15.28	3	422.6	-15.28	6.2
M-RC206	171.1	421.3	146.23	2	10800	367.3	-12.82	2	367.3	-12.82	6.3
M-RC207	145.1	370.2	155.13	2	10800	309.9	-16.29	2	309.9	-16.29	6.7
M-RC208	144.1	289.5	100.90	1	10800	289.5	0.00	1	289.5	0.00	8.6
Average	221.8	348.6	64.94	2.9	10800	380.2	-1.85	3.2	380.2	-1.83	6.3

Table D.6: ALNS results for 25-node problems using Category 6 parameters

Instances	CPLEX					The Proposed ALNS					
	lower bound	upper bound	gap (%)	#veh.	sol. time (secs)	best dist.	best gap (%)	best #veh.	avg. dist.	avg. gap (%)	avg. time (secs)
M-R101	451.1	612.3	35.73	7	10800	612.3	0.00	7	612.3	0.00	7.1
M-R102	356.1	590.9	65.94	7	10800	578.2	-2.15	7	578.2	-2.15	7.1
M-R103	-	-	-	-	-	517.9	-	5	517.9	-	7.1
M-R104	-	-	-	-	-	500.3	-	5	500.3	-	9.3
M-R105	399.8	512.4	28.16	6	10800	512.4	0.00	6	512.4	0.00	6.9
M-R106	309.7	489.8	58.15	4	10800	477.4	-2.53	4	477.4	-2.53	7
M-R107	-	-	-	-	-	499.3	-	4	499.3	-	7.3
M-R108	-	-	-	-	-	449.7	-	4	449.7	-	7
M-R109	-	-	-	-	-	563.5	-	6	563.5	-	7.2
M-R110	-	-	-	-	-	564.5	-	5	564.5	-	6.6
M-R111	-	-	-	-	-	462.5	-	5	462.5	-	7.1
M-R112	-	-	-	-	-	690.5	-	5	691.7	-	9.1
M-C101	179.4	350.2	95.21	5	10800	335.2	-4.28	4	335.2	-4.28	7
M-C102	-	-	-	-	-	267.0	-	3	267.0	-	6.2
M-C103	96.4	199.3	-	3	10800	199.3	0.00	3	199.3	0.00	7
M-C104	128.5	197.9	54.01	3	10800	197.9	0.00	3	197.9	0.00	7.4
M-C105	183.8	320.5	74.37	4	10800	320.5	0.00	4	320.5	0.00	8.1
M-C106	182.5	307.5	68.49	5	10800	307.5	0.00	5	307.5	0.00	6.8
M-C107	160.3	289.3	80.47	4	10800	281.9	-2.56	4	281.9	-2.56	7.8
M-C108	-	-	-	-	-	317.6	-	4	317.6	-	7.7
M-C109	-	-	-	-	-	285.8	-	3	285.8	-	8.1
M-RC101	219.4	535.6	144.12	5	10800	535.6	0.00	5	535.6	0.00	6.4
M-RC102	-	-	-	-	-	584.9	-	5	584.9	-	6.8
M-RC103	-	-	-	-	-	390.0	-	3	390.0	-	7.2
M-RC104	-	-	-	-	-	432.0	-	3	432.0	-	7.4
M-RC105	-	-	-	-	-	592.7	-	5	592.7	-	6.8
M-RC106	-	-	-	-	-	631.3	-	5	631.3	-	6.2
M-RC107	-	-	-	-	-	446.0	-	4	446.0	-	6.6
M-RC108	-	-	-	-	-	494.0	-	4	494.0	-	7.8
M-R201	359.4	421.1	17.17	3	10800	421.1	0.00	3	421.1	0.00	7.1
M-R202	279.2	422.5	51.33	3	10800	379.0	-10.30	2	379.0	-10.30	6.9
M-R203	294.6	379.9	28.95	3	10800	372.8	-1.87	2	372.8	-1.87	7.5
M-R204	274.3	355.8	29.71	3	10800	354.0	-0.51	2	354.3	-0.42	7.2
M-R205	311.4	386.0	23.96	3	10800	386.0	0.00	3	386.0	0.00	8.1
M-R206	274.4	365.3	33.13	3	10800	365.3	0.00	3	365.3	0.00	7
M-R207	275.9	360.6	30.70	2	10800	352.7	-2.19	2	352.7	-2.19	7.2
M-R208	277.7	304.3	9.58	1	10800	303.6	-0.23	1	303.6	-0.23	7.1
M-R209	306.1	420.6	37.41	2	10800	401.3	-4.59	2	401.6	-4.52	8.2
M-R210	297.8	432.5	45.23	2	10800	430.0	-0.58	2	430.0	-0.58	7.5
M-R211	267.0	371.1	38.99	3	10800	341.9	-7.87	2	341.9	-7.87	7
M-C201	211.0	246.8	16.97	2	10800	246.8	0.00	2	246.8	0.00	7.8
M-C202	178.5	213.5	19.61	2	10800	213.5	0.00	2	213.5	0.00	8.8
M-C203	163.6	253.1	54.71	1	10800	232.0	-8.34	1	232.0	-8.34	9.4
M-C204	153.0	202.1	32.09	1	10800	202.1	0.00	1	202.1	0.00	8
M-C205	195.5	264.7	35.40	2	10800	264.7	0.00	2	264.7	0.00	6.9
M-C206	173.3	258.4	49.11	2	10800	258.4	0.00	2	258.4	0.00	6.5
M-C207	157.5	252.4	60.25	2	10800	236.1	-6.46	2	236.1	-6.46	7.5
M-C208	166.3	259.8	56.22	2	10800	252.4	-2.85	1	252.4	-2.85	6.6
M-RC201	184.1	438.8	138.35	3	10800	425.1	-3.12	3	425.1	-3.12	6.8
M-RC202	154.2	394.4	155.77	2	10800	394.1	-0.08	2	394.1	-0.08	8
M-RC203	160.4	424.4	164.59	2	10800	376.8	-11.22	2	389.2	-8.29	7.9
M-RC204	155.0	328.8	112.13	2	10800	306.8	-6.69	1	306.8	-6.69	7.4
M-RC205	177.7	503.2	183.17	2	10800	419.6	-16.61	3	420.1	-16.51	7.5
M-RC206	172.4	374.5	117.23	2	10800	367.3	-1.92	2	367.3	-1.92	7
M-RC207	146.3	344.7	135.61	3	10800	344.7	0.00	3	344.7	0.00	7.2
M-RC208	142.9	289.5	102.59	1	10800	288.8	-0.24	1	289.4	-0.03	9.9
Average	225.7	359.9	67.15	2.9	10800	392.6	-2.56	3.3	392.9	-2.47	7.4

Table D.7: ALNS results for 25-node problems using Category 7 parameters

Instances	CPLEX lower bound	upper bound	gap (%)	#veh.	sol. time (secs)	The Proposed ALNS best dist.	best gap (%)	best #veh.	avg. dist.	avg. gap (%)	avg. time (secs)
M-R101	621.5	748.2	20.39	9	10800	748.2	0.00	9	748.2	0.00	6.5
M-R102	-	-	-	-	-	578.5	-	7	578.5	-	6.9
M-R103	-	-	-	-	-	537.2	-	5	537.2	-	5.7
M-R104	315.7	464.9	47.26	5	10800	449.8	-3.25	4	449.8	-3.25	5.8
M-R105	-	-	-	-	-	916.6	-	9	916.6	-	8.7
M-R106	-	-	-	-	-	803.6	-	7	803.6	-	6.8
M-R107	-	-	-	-	-	449.9	-	4	449.9	-	5.7
M-R108	-	-	-	-	-	486.3	-	4	486.3	-	6.5
M-R109	-	-	-	-	-	592.6	-	5	592.6	-	6.3
M-R110	-	-	-	-	-	608.9	-	5	608.9	-	6.1
M-R111	313.2	532.9	70.15	5	10800	492.3	-7.62	5	492.3	-7.62	6.1
M-R112	-	-	-	-	-	452.9	-	4	452.9	-	6.1
M-C101	191.9	310.3	61.70	4	10800	310.3	0.00	4	310.3	0.00	6.5
M-C102	147.7	297.1	101.15	4	10800	286.1	-3.70	3	286.1	-3.70	6.3
M-C103	107.0	201.4	88.22	3	10800	201.4	0.00	3	201.4	0.00	6.2
M-C104	-	-	-	-	-	259.0	-	3	259.1	-	7.1
M-C105	287.2	385.7	34.30	5	10800	385.7	0.00	5	385.7	0.00	7
M-C106	223.6	339.5	51.83	5	10800	339.5	0.00	5	339.5	0.00	6.9
M-C107	174.2	270.0	54.99	4	10800	270.0	0.00	4	270.0	0.00	6
M-C108	-	-	-	-	-	483.1	-	5	483.1	-	9
M-C109	104.3	293.7	181.59	3	10800	293.7	0.00	3	293.7	0.00	6
M-RC101	322.3	546.9	69.69	5	10800	546.9	0.00	5	546.9	0.00	6.3
M-RC102	-	-	-	-	-	604.1	-	5	604.1	-	7.1
M-RC103	-	-	-	-	-	460.1	-	4	460.1	-	6.5
M-RC104	-	-	-	-	-	699.5	-	5	699.5	-	6.4
M-RC105	-	-	-	-	-	751.9	-	6	751.9	-	6.9
M-RC106	-	-	-	-	-	601.6	-	5	601.6	-	6.3
M-RC107	-	-	-	-	-	456.7	-	4	456.7	-	6.5
M-RC108	-	-	-	-	-	607.5	-	4	608.9	-	7.7
M-R201	405.9	514.9	26.85	3	10800	514.9	0.00	3	514.9	0.00	5.8
M-R202	305.4	480.6	57.37	3	10800	453.1	-5.72	3	453.1	-5.72	6
M-R203	304.9	401.3	31.62	3	10800	396.1	-1.30	2	396.1	-1.30	6
M-R204	282.2	399.3	41.50	2	10800	388.9	-2.60	2	388.9	-2.60	5.7
M-R205	338.8	411.2	21.37	2	10800	411.2	0.00	2	411.2	0.00	6.2
M-R206	296.4	401.5	35.46	1	10800	401.3	-0.05	1	401.3	-0.05	6.7
M-R207	291.9	401.6	37.58	1	10800	392.9	-2.17	1	393.6	-1.99	7.1
M-R208	290.2	353.3	21.74	2	10800	333.8	-5.52	1	334.3	-5.38	7.3
M-R209	318.3	431.0	35.41	3	10800	425.0	-1.39	2	425.0	-1.39	5.9
M-R210	313.3	503.6	60.74	3	10800	480.1	-4.67	3	480.2	-4.65	7.2
M-R211	295.7	375.4	26.95	2	10800	362.4	-3.46	1	364.4	-2.93	6.7
M-C201	226.4	226.4	0.00	2	152	226.4	0.00	2	226.4	0.00	6.7
M-C202	216.5	266.5	23.09	2	10800	266.5	0.00	2	266.5	0.00	7.2
M-C203	169.1	265.7	57.13	1	10800	251.2	-5.46	1	251.2	-5.46	8
M-C204	161.2	238.9	48.20	1	10800	238.9	0.00	1	238.9	0.00	7.2
M-C205	202.3	244.7	20.96	2	10800	244.7	0.00	2	244.7	0.00	7
M-C206	201.2	303.8	50.99	2	10800	293.6	-3.36	2	293.6	-3.36	7.3
M-C207	213.3	266.7	25.04	2	10800	265.0	-0.64	2	265.0	-0.64	6.1
M-C208	179.4	227.3	26.70	1	10800	227.3	0.00	1	227.3	0.00	6.9
M-RC201	282.0	458.5	62.59	3	10800	458.5	0.00	3	458.5	0.00	7.2
M-RC202	175.1	475.5	171.56	3	10800	413.3	-13.08	2	413.3	-13.08	6.2
M-RC203	166.4	376.3	126.14	3	10800	353.6	-6.03	2	353.6	-6.03	6.1
M-RC204	157.2	389.3	147.65	1	10800	389.0	-0.08	1	389.0	-0.08	6.7
M-RC205	192.2	516.9	168.94	3	10800	431.3	-16.56	3	431.3	-16.56	5.8
M-RC206	191.6	474.8	147.81	2	10800	388.1	-18.26	2	388.1	-18.26	6.4
M-RC207	158.8	391.8	146.73	3	10800	349.2	-10.87	2	349.2	-10.87	6.3
M-RC208	146.4	290.2	98.22	2	10800	286.3	-1.34	1	288.3	-0.65	6.1
Average	244.5	381.0	65.78	2.9	10520	434.2	-3.08	3.4	434.3	-3.04	6.6

Table D.8: ALNS results for 25-node problems using Category 8 parameters

Instances	CPLEX lower bound	upper bound	gap (%)	#veh.	sol. time (secs)	The Proposed ALNS best dist.	best gap (%)	best #veh.	avg. dist.	avg. gap (%)	avg. time (secs)
M-R101	451.5	625.8	38.60	8	10800	625.8	0.00	8	625.8	0.00	5.3
M-R102	335.5	547.9	63.31	7	10800	547.9	0.00	7	547.9	0.00	4.7
M-R103	-	-	-	-	-	470.0	-	5	470.0	-	5.4
M-R104	298.1	434.2	45.66	4	10800	430.6	-0.83	4	430.6	-0.83	4.9
M-R105	399.3	503.4	26.07	6	10800	503.4	0.00	6	503.4	0.00	4.5
M-R106	316.0	468.3	48.20	5	10800	459.2	-1.94	4	459.2	-1.94	5
M-R107	286.0	403.3	41.01	4	10800	403.3	0.00	4	403.3	0.00	4.5
M-R108	-	-	-	-	-	398.2	-	3	398.2	-	4.2
M-R109	-	-	-	-	-	495.8	-	5	495.8	-	4.8
M-R110	-	-	-	-	-	577.4	-	5	577.4	-	4.6
M-R111	299.7	430.0	43.48	4	10800	430.0	0.00	4	430.0	0.00	5.3
M-R112	-	-	-	-	-	537.7	-	4	537.7	-	6.2
M-C101	182.2	273.2	49.95	4	10800	273.2	0.00	4	273.2	0.00	5.3
M-C102	147.1	274.1	86.34	4	10800	267.0	-2.59	3	267.0	-2.59	4.2
M-C103	109.5	186.1	69.95	3	10800	186.1	0.00	3	186.1	0.00	5.3
M-C104	132.9	193.0	45.22	3	10800	179.6	-6.94	2	179.6	-6.94	4.7
M-C105	170.0	277.1	63.00	4	10800	277.1	0.00	4	277.1	0.00	5.4
M-C106	216.6	281.9	30.15	5	10800	281.9	0.00	5	281.9	0.00	4.9
M-C107	152.8	231.9	51.77	3	10800	231.9	0.00	3	231.9	0.00	4.5
M-C108	123.6	227.9	84.39	3	10800	227.9	0.00	3	227.9	0.00	5
M-C109	105.3	235.8	123.93	3	10801	225.7	-4.28	3	225.7	-4.28	4.4
M-RC101	243.4	478.4	96.55	4	10800	478.4	0.00	4	478.4	0.00	4.2
M-RC102	-	-	-	-	-	455.9	-	4	455.9	-	4.8
M-RC103	-	-	-	-	-	384.7	-	3	384.7	-	4.6
M-RC104	-	-	-	-	-	385.5	-	3	385.5	-	5.3
M-RC105	-	-	-	-	-	423.5	-	4	423.5	-	4.3
M-RC106	-	-	-	-	-	436.8	-	4	436.8	-	5.1
M-RC107	-	-	-	-	-	364.9	-	3	364.9	-	4.4
M-RC108	-	-	-	-	-	448.6	-	4	448.6	-	4.6
M-R201	356.5	414.8	16.35	3	10800	414.8	0.00	3	414.8	0.00	4.7
M-R202	267.3	359.3	34.42	2	10800	359.3	0.00	2	359.3	0.00	5.4
M-R203	289.0	357.5	23.70	3	10800	351.1	-1.79	2	351.1	-1.79	4.9
M-R204	275.5	354.0	28.49	2	10800	345.6	-2.37	2	345.6	-2.37	4.5
M-R205	311.7	382.9	22.84	3	10800	382.9	0.00	3	382.9	0.00	5
M-R206	265.4	346.1	30.41	2	10800	346.1	0.00	2	346.1	0.00	4.5
M-R207	275.3	357.7	29.93	2	10800	339.7	-5.03	3	339.7	-5.03	4.2
M-R208	265.0	303.6	14.57	1	10800	303.6	0.00	1	303.7	0.03	6.3
M-R209	299.6	390.5	30.34	2	10800	386.0	-1.15	2	386.0	-1.15	5.2
M-R210	293.8	391.1	33.12	2	10800	388.0	-0.79	2	388.0	-0.79	5.1
M-R211	271.3	344.8	27.09	2	10800	335.2	-2.78	2	335.2	-2.78	6.2
M-C201	229.5	239.2	4.23	2	10800	239.2	0.00	2	239.2	0.00	4.9
M-C202	176.3	213.5	21.10	2	10800	213.5	0.00	2	213.5	0.00	5.1
M-C203	170.6	227.2	33.18	1	10800	227.2	0.00	1	227.2	0.00	6.3
M-C204	154.1	202.1	31.15	1	10800	202.1	0.00	1	202.1	0.00	5.2
M-C205	209.5	264.7	26.35	2	10800	264.7	0.00	2	264.7	0.00	5.1
M-C206	175.7	253.8	44.45	2	10800	241.8	-4.73	2	241.8	-4.73	4.7
M-C207	157.6	228.6	45.05	2	10800	228.6	0.00	2	228.6	0.00	4.6
M-C208	168.4	212.3	26.07	2	10800	212.3	0.00	2	212.3	0.00	5.1
M-RC201	201.5	424.3	110.57	3	10800	368.8	-13.08	3	368.8	-13.08	5.1
M-RC202	158.7	388.4	144.74	2	10800	388.4	0.00	2	388.4	0.00	5.2
M-RC203	155.7	342.0	119.65	2	10800	321.0	-6.14	3	321.0	-6.14	6.1
M-RC204	155.2	260.8	68.04	2	10800	260.8	0.00	2	260.8	0.00	4.8
M-RC205	181.3	445.3	145.62	3	10800	415.9	-6.60	3	415.9	-6.60	4.4
M-RC206	171.0	360.3	110.70	2	10800	354.8	-1.53	3	354.8	-1.53	5.1
M-RC207	144.9	326.2	125.12	2	10800	309.6	-5.09	2	309.6	-5.09	5
M-RC208	145.0	290.4	100.28	1	10800	289.5	-0.31	1	289.5	-0.31	6.7
Average	224.9	335.3	55.80	2.9	10800	355.3	-1.55	3.1	355.3	-1.54	5.0

Table D.9: ALNS results for 25-node problems using Category 9 parameters

Instances	CPLEX					The Proposed ALNS					
	lower bound	upper bound	gap (%)	#veh.	sol. time (secs)	best dist.	best gap (%)	best #veh.	avg. dist.	avg. gap (%)	avg. time (secs)
M-R101	559.0	693.0	23.97	9	10800	693.0	0.00	9	693.0	0.00	5.4
M-R102	363.5	564.4	55.27	7	10800	564.4	0.00	7	564.4	0.00	6
M-R103	-	-	-	-	-	551.9	-	5	551.9	-	5.3
M-R104	-	-	-	-	-	465.6	-	4	465.6	-	5.9
M-R105	422.0	555.0	31.52	6	10800	555.0	0.00	6	555.0	0.00	5.6
M-R106	311.6	501.8	61.04	5	10800	498.8	-0.60	5	498.8	-0.60	5.1
M-R107	287.3	418.7	45.74	4	10800	418.7	0.00	4	418.7	0.00	5
M-R108	-	-	-	-	-	419.8	-	3	419.8	-	4.8
M-R109	-	-	-	-	-	478.7	-	5	478.7	-	5.1
M-R110	-	-	-	-	-	580.2	-	5	580.2	-	5.4
M-R111	309.6	487.5	57.46	5	10800	455.0	-6.67	4	455.0	-6.67	5
M-R112	-	-	-	-	-	484.7	-	4	484.7	-	5.7
M-C101	184.6	280.9	52.17	4	10800	280.9	0.00	4	280.9	0.00	5.9
M-C102	151.7	290.5	91.50	4	10800	255.0	-12.22	3	255.0	-12.22	4.9
M-C103	104.0	200.2	92.50	3	10800	198.0	-1.10	3	198.0	-1.10	5.6
M-C104	135.8	244.7	80.19	3	10800	244.7	0.00	3	244.7	0.00	5.7
M-C105	186.7	288.9	54.74	4	10800	288.9	0.00	4	288.9	0.00	5.5
M-C106	230.4	358.6	55.64	6	10800	358.6	0.00	6	358.6	0.00	5.6
M-C107	153.5	257.5	67.75	4	10800	257.5	0.00	4	257.5	0.00	5.9
M-C108	129.9	272.4	109.70	3	10800	261.4	-4.04	3	261.4	-4.04	5.5
M-C109	108.5	249.6	130.05	3	10800	241.3	-3.33	3	241.3	-3.33	5.2
M-RC101	291.1	587.0	101.65	5	10800	587.0	0.00	5	587.0	0.00	4.6
M-RC102	-	-	-	-	-	450.9	-	4	450.9	-	5.2
M-RC103	-	-	-	-	-	384.9	-	3	384.9	-	4.4
M-RC104	-	-	-	-	-	488.4	-	4	488.4	-	5.9
M-RC105	-	-	-	-	-	531.7	-	4	531.7	-	4.6
M-RC106	-	-	-	-	-	503.0	-	4	503.0	-	5.4
M-RC107	188.8	472.7	150.37	4	10800	403.3	-14.68	3	403.5	-14.64	5.2
M-RC108	-	-	-	-	-	504.0	-	4	504.0	-	5.4
M-R201	377.8	444.1	17.55	3	10800	444.1	0.00	3	444.1	0.00	5.3
M-R202	288.0	393.0	36.46	2	10800	393.0	0.00	2	393.0	0.00	6
M-R203	296.8	390.8	31.67	3	10800	382.4	-2.15	2	382.4	-2.15	5.4
M-R204	287.2	353.7	23.15	2	10800	351.1	-0.74	2	351.1	-0.74	5
M-R205	323.2	406.1	25.65	3	10800	395.4	-2.63	3	395.4	-2.63	5.1
M-R206	285.5	362.4	26.94	2	10800	362.4	0.00	2	362.4	0.00	5.4
M-R207	287.4	355.7	23.76	2	10800	355.7	0.00	2	355.7	0.00	5.1
M-R208	283.5	325.2	14.71	1	10800	316.9	-2.55	1	316.9	-2.55	6.2
M-R209	302.9	391.0	29.09	2	10800	385.2	-1.48	2	385.2	-1.48	5.4
M-R210	294.8	426.7	44.74	2	10800	426.7	0.00	2	426.7	0.00	5.6
M-R211	283.0	364.8	28.90	2	10800	356.3	-2.33	1	356.3	-2.33	6.2
M-C201	211.4	211.4	0.00	2	40	211.4	0.00	2	211.4	0.00	5.6
M-C202	177.5	213.5	20.28	2	10800	213.5	0.00	2	213.5	0.00	5.7
M-C203	163.1	238.9	46.47	2	10800	238.9	0.00	2	238.9	0.00	5.7
M-C204	159.3	231.2	45.13	2	10800	229.3	-0.82	1	229.3	-0.82	6.4
M-C205	215.8	292.7	35.63	2	10800	292.7	0.00	2	292.7	0.00	5.9
M-C206	177.7	236.6	33.15	2	10800	236.6	0.00	2	236.6	0.00	5.6
M-C207	164.0	219.5	33.84	2	10800	219.5	0.00	2	219.5	0.00	5.4
M-C208	169.3	259.3	53.16	2	10800	251.2	-3.12	1	251.2	-3.12	5.5
M-RC201	234.8	451.0	92.08	3	10800	441.4	-2.13	3	441.4	-2.13	5.3
M-RC202	167.6	409.6	144.39	2	10800	386.5	-5.64	2	386.5	-5.64	5.2
M-RC203	161.4	342.8	112.39	2	10800	342.8	0.00	2	342.8	0.00	4.8
M-RC204	158.0	300.6	90.25	2	10800	300.6	0.00	2	300.6	0.00	5.5
M-RC205	180.2	406.4	125.53	2	10800	385.6	-5.12	2	392.8	-3.35	5.3
M-RC206	179.1	442.2	146.90	3	10800	418.3	-5.40	2	418.3	-5.40	5.9
M-RC207	149.9	400.7	167.31	2	10800	398.5	-0.55	2	398.5	-0.55	5.2
M-RC208	146.6	295.1	101.30	2	10800	266.7	-9.62	1	269.6	-8.64	5.4
Average	232.8	361.1	63.90	3.1	10555	382.3	-1.98	3.2	382.5	-1.91	5.4

Table D.10: ALNS results for 25-node problems using Category 10 parameters

Instances	CPLEX lower bound	upper bound	gap (%)	#veh.	sol. time (secs)	The Proposed ALNS best dist.	best gap (%)	best #veh.	avg. dist.	avg. gap (%)	avg. time (secs)
M-R101	416.9	600.8	44.11	7	10800	600.8	0.00	7	600.8	0.00	12.1
M-R102	-	-	-	-	-	528.6	-	7	529.0	-	12.2
M-R103	-	-	-	-	-	722.4	-	6	722.4	-	12.9
M-R104	-	-	-	-	-	472.6	-	4	474.9	-	11.6
M-R105	375.5	526.7	40.27	6	10800	526.7	0.00	6	526.7	0.00	11
M-R106	-	-	-	-	-	500.5	-	4	500.5	-	10.6
M-R107	-	-	-	-	-	423.1	-	4	423.1	-	9.3
M-R108	-	-	-	-	-	475.9	-	4	475.9	-	12.4
M-R109	-	-	-	-	-	490.2	-	5	490.2	-	10.2
M-R110	264.3	461.4	74.57	4	10800	453.5	-1.71	4	453.5	-1.71	10.4
M-R111	-	-	-	-	-	458.5	-	5	459.0	-	10.7
M-R112	-	-	-	-	-	445.4	-	4	445.4	-	10.4
M-C101	233.0	294.3	26.31	4	10800	294.3	0.00	4	294.3	0.00	13.4
M-C102	178.5	364.4	104.15	5	2228	227.1	-37.68	3	227.1	-37.68	12.9
M-C103	95.7	197.5	106.37	3	10800	181.4	-8.15	2	181.4	-8.15	12.7
M-C104	121.5	220.1	81.15	3	10800	202.7	-7.91	3	202.7	-7.91	16
M-C105	184.9	203.5	10.06	3	10800	203.5	0.00	3	203.5	0.00	10.2
M-C106	192.8	226.4	17.43	3	7893	226.4	0.00	3	226.4	0.00	9.7
M-C107	147.3	187.8	27.49	3	5469	187.8	0.00	3	187.8	0.00	10.8
M-C108	-	-	-	-	-	278.5	-	4	279.8	-	12.7
M-C109	-	-	-	-	-	277.5	-	4	277.6	-	13.3
M-RC101	-	-	-	-	-	723.4	-	6	723.4	-	10
M-RC102	-	-	-	-	-	352.7	-	3	352.7	-	12.7
M-RC103	-	-	-	-	-	345.4	-	3	345.4	-	10.1
M-RC104	-	-	-	-	-	411.1	-	3	411.1	-	11.5
M-RC105	-	-	-	-	-	751.3	-	6	751.3	-	12.4
M-RC106	-	-	-	-	-	459.5	-	4	459.5	-	10.6
M-RC107	-	-	-	-	-	394.8	-	3	400.2	-	10.7
M-RC108	-	-	-	-	-	298.8	-	3	298.8	-	11.6
M-R201	325.4	424.5	30.45	2	10800	420.3	-0.99	3	420.3	-0.99	10.9
M-R202	256.2	357.8	39.66	2	10800	347.6	-2.85	2	347.6	-2.85	10.2
M-R203	284.1	340.0	19.68	2	10800	340.0	0.00	2	340.0	0.00	10.6
M-R204	260.4	334.3	28.38	2	10800	334.3	0.00	2	334.3	0.00	10.9
M-R205	296.0	402.3	35.91	2	10800	384.9	-4.33	2	385.6	-4.15	12.2
M-R206	255.2	352.3	38.05	2	10800	331.1	-6.02	2	331.1	-6.02	11.2
M-R207	268.3	340.7	26.98	2	10800	327.4	-3.90	2	329.9	-3.17	11.3
M-R208	259.4	296.2	14.19	1	10800	296.2	0.00	1	296.8	0.20	10.8
M-R209	292.7	368.1	25.76	2	10800	368.1	0.00	2	368.1	0.00	11.6
M-R210	281.1	380.9	35.50	2	10800	380.9	0.00	2	380.9	0.00	11.4
M-R211	255.6	347.5	35.95	1	10800	327.9	-5.64	1	332.1	-4.43	12.3
M-C201	239.3	254.7	6.44	2	10800	254.7	0.00	2	254.7	0.00	10.8
M-C202	179.5	289.2	61.11	2	4305	243.7	-15.73	1	243.7	-15.73	13.3
M-C203	172.8	325.0	88.08	2	2486	226.9	-30.18	1	226.9	-30.18	13.2
M-C204	166.6	201.8	21.13	1	10800	201.8	0.00	1	201.8	0.00	12
M-C205	229.9	273.8	19.10	2	5838	273.8	0.00	2	273.8	0.00	12.6
M-C206	186.6	270.0	44.69	2	4768	245.3	-9.15	2	245.3	-9.15	11.6
M-C207	199.3	210.4	5.57	2	10800	210.4	0.00	2	210.4	0.00	11.5
M-C208	206.5	255.2	23.58	2	6642	255.2	0.00	2	255.2	0.00	13.2
M-RC201	168.5	362.4	115.07	3	10800	362.4	0.00	3	362.4	0.00	10.6
M-RC202	159.8	401.2	151.06	2	10800	371.6	-7.38	2	371.6	-7.38	11.9
M-RC203	150.9	522.5	246.26	2	10800	340.0	-34.93	2	340.0	-34.93	12.9
M-RC204	144.5	305.5	111.42	1	10800	295.6	-3.24	2	295.6	-3.24	11.7
M-RC205	172.5	462.9	168.35	2	10800	434.3	-6.18	2	434.3	-6.18	11.6
M-RC206	163.7	460.4	181.25	2	10800	372.9	-19.01	2	372.9	-19.01	11.9
M-RC207	145.9	336.6	130.71	2	10800	336.6	0.00	2	336.6	0.00	12.1
M-RC208	138.1	252.8	83.06	1	10800	249.2	-1.42	1	249.6	-1.27	12.3
Average	218.1	335.5	62.68	2.5	9536	365.1	-5.58	3.0	365.5	-5.51	11.6

Table D.11: ALNS results for 25-node problems using Category 11 parameters

Instances	CPLEX lower bound	upper bound	gap (%)	#veh.	sol. time (secs)	The Proposed ALNS best dist.	best gap (%)	best #veh.	avg. dist.	avg. gap (%)	avg. time (secs)
M-R101	640.5	706.6	10.32	9	10800	706.6	0.00	9	706.6	0.00	5.1
M-R102	367.4	563.4	53.35	7	10800	555.9	-1.33	7	555.9	-1.33	5.9
M-R103	316.6	475.1	50.06	5	10800	475.1	0.00	5	475.1	0.00	4.7
M-R104	314.0	437.8	39.43	4	10800	432.7	-1.16	4	432.7	-1.16	5.1
M-R105	447.2	594.2	32.87	7	10800	594.2	0.00	7	594.2	0.00	5.5
M-R106	335.4	508.5	51.61	5	10800	481.7	-5.27	5	481.7	-5.27	4.6
M-R107	304.7	434.1	42.47	4	10800	434.1	0.00	4	434.1	0.00	4.8
M-R108	289.2	419.2	44.95	4	10800	416.4	-0.67	4	416.4	-0.67	4.5
M-R109	331.6	484.5	46.11	5	10800	481.4	-0.64	5	481.4	-0.64	4.9
M-R110	313.3	486.6	55.31	5	10800	486.0	-0.12	5	486.0	-0.12	4.8
M-R111	311.4	430.0	38.09	4	10800	419.9	-2.35	4	419.9	-2.35	4.1
M-R112	-	-	-	-	-	409.6	-	4	409.6	-	4.4
M-C101	191.3	191.3	0.00	3	5	191.3	0.00	3	191.3	0.00	5
M-C102	163.9	191.6	16.90	3	10800	191.6	0.00	3	191.6	0.00	4.5
M-C103	144.3	199.9	38.53	3	10800	199.9	0.00	3	199.9	0.00	4.7
M-C104	145.3	219.2	50.86	3	10800	219.2	0.00	3	219.2	0.00	5.2
M-C105	231.9	231.9	0.00	3	75	231.9	0.00	3	231.9	0.00	4.5
M-C106	234.4	234.4	0.00	4	134	234.4	0.00	4	234.4	0.00	4.4
M-C107	230.7	269.6	16.86	4	10800	269.6	0.00	4	269.6	0.00	4.4
M-C108	164.6	193.2	17.38	3	10800	193.2	0.00	3	193.2	0.00	5
M-C109	120.4	196.7	63.37	3	10800	196.7	0.00	3	196.7	0.00	4.8
M-RC101	340.0	363.6	6.94	3	10800	363.6	0.00	3	363.6	0.00	4.5
M-RC102	288.0	431.4	49.79	4	10800	431.4	0.00	4	431.4	0.00	4.9
M-RC103	269.6	368.8	36.80	3	10800	368.8	0.00	3	368.8	0.00	4.7
M-RC104	218.4	397.6	82.05	4	10800	397.6	0.00	4	397.6	0.00	4.7
M-RC105	245.6	538.2	119.14	5	10800	510.7	-5.11	5	510.7	-5.11	4.4
M-RC106	302.3	440.4	45.68	4	10800	440.4	0.00	4	440.4	0.00	4.5
M-RC107	266.1	396.2	48.89	4	10800	372.5	-5.98	3	372.5	-5.98	4.8
M-RC108	272.4	304.1	11.64	3	10800	304.1	0.00	3	304.1	0.00	4.4
M-R201	391.7	454.5	16.03	3	10800	454.4	-0.02	4	454.4	-0.02	5.4
M-R202	302.4	406.1	34.29	3	10800	391.6	-3.57	3	391.6	-3.57	4.9
M-R203	302.9	389.5	28.59	2	10800	384.2	-1.36	2	384.2	-1.36	4.7
M-R204	286.0	362.2	26.64	3	10800	357.8	-1.21	2	357.8	-1.21	4.6
M-R205	340.5	381.9	12.16	3	10800	381.9	0.00	3	381.9	0.00	4.6
M-R206	283.1	383.1	35.32	1	10800	377.8	-1.38	2	377.8	-1.38	6
M-R207	289.0	352.4	21.94	2	10800	350.6	-0.51	2	350.6	-0.51	5.5
M-R208	291.3	326.1	11.95	1	10800	326.1	0.00	1	326.2	0.03	5.9
M-R209	301.7	396.9	31.55	3	10800	380.4	-4.16	3	380.4	-4.16	4.8
M-R210	287.7	446.7	55.27	2	10800	410.5	-8.10	2	410.5	-8.10	5.7
M-R211	287.0	351.4	22.44	2	10800	351.4	0.00	2	351.4	0.00	6
M-C201	208.2	208.2	0.00	2	5	208.2	0.00	2	208.2	0.00	4.4
M-C202	256.0	256.0	0.00	2	1618	256.0	0.00	2	256.0	0.00	5.1
M-C203	177.3	239.4	35.03	2	10800	239.4	0.00	2	239.4	0.00	5.2
M-C204	171.2	216.5	26.46	2	10800	213.3	-1.48	1	213.8	-1.25	5.8
M-C205	244.7	244.7	0.00	2	3039	244.7	0.00	2	244.7	0.00	5.3
M-C206	228.4	228.4	0.00	2	4973	228.4	0.00	2	228.4	0.00	4.9
M-C207	231.3	237.7	2.77	2	10800	237.7	0.00	2	237.7	0.00	5.3
M-C208	210.5	214.5	1.90	2	10800	214.5	0.00	2	214.5	0.00	5.1
M-RC201	258.5	452.0	74.85	2	10800	442.5	-2.10	3	442.5	-2.10	5.1
M-RC202	181.7	331.2	82.28	3	10800	331.2	0.00	3	331.2	0.00	4.5
M-RC203	165.1	326.2	97.58	3	10800	326.2	0.00	3	326.2	0.00	5.8
M-RC204	162.0	325.9	101.17	1	10800	321.0	-1.50	2	321.0	-1.50	5.5
M-RC205	209.6	326.1	55.58	3	10800	326.1	0.00	3	326.1	0.00	5.3
M-RC206	199.7	359.7	80.12	2	10800	359.7	0.00	2	359.7	0.00	5.2
M-RC207	163.4	305.8	87.15	2	10800	305.8	0.00	2	305.8	0.00	4.9
M-RC208	154.0	263.4	71.04	2	10800	263.4	0.00	2	263.4	0.00	6.1
Average	261.6	354.4	37.85	3.2	9605	351.7	-0.87	3.3	351.7	-0.87	5.0

Appendix E: Detailed ALNS results of new benchmark instances with 50 nodes

We report here the detailed ALNS results of new benchmark instances with 50-nodes.

Table E.1: ALNS results for 50-node problems using Category 1, 2 and 3 parameters

The Proposed ALNS Results for Instances with 50 Nodes												
	Category 1				Category 2				Category 3			
Instances	best dist.	best #veh.	avg. dist.	avg. time (secs)	best dist.	best #veh.	avg. dist.	avg. time (secs)	best dist.	best #veh.	avg. dist.	avg. time (secs)
M-R101	1324.5	14	1325.4	21.8	1408.1	15	1409.8	23.6	1304.7	14	1304.7	21.1
M-R102	899.9	10	899.9	19.8	894.7	10	894.7	19.7	902.8	10	902.9	20.4
M-R103	907.1	9	907.1	17.1	898.5	9	899.3	17.4	891.6	9	891.6	17
M-R104	652.3	6	652.7	16.9	740.8	6	740.8	18.2	710.4	6	710.4	17.8
M-R105	830.7	9	830.7	17.7	944.3	10	944.3	18.9	979.7	10	979.7	20.2
M-R106	985.2	10	985.3	19.5	997.1	10	997.1	20.7	1004.4	9	1004.4	21.8
M-R107	738.8	7	740.9	17.6	729.8	7	729.8	17.6	749.7	8	749.7	16.9
M-R108	623.7	6	623.7	18.1	623.7	6	623.8	40.8	623.7	6	623.7	18.2
M-R109	878.2	8	882.3	17	856.7	8	856.7	18.3	892.1	8	892.9	17.2
M-R110	673.2	7	674.2	17	1068.7	9	1069.7	19.1	922.8	8	922.8	16.9
M-R111	786.1	7	786.1	16.3	863.5	7	864.4	18.3	849.5	7	849.5	16
M-R112	656.7	6	657.2	16.8	687.5	6	694.2	18.5	694.1	6	701.7	19.1
M-C101	521.5	7	521.5	18	539.2	7	539.2	18.6	602.5	7	602.8	19.7
M-C102	481.1	6	483.1	19.5	497.3	6	497.3	20.2	576.0	7	576.0	21.4
M-C103	454.7	5	454.8	22	462.5	5	462.5	21.4	531.2	6	531.2	21.5
M-C104	447.4	5	448.8	18.4	415.7	5	415.7	19.6	464.3	5	464.3	18.5
M-C105	453.8	6	453.8	17.7	453.8	6	453.8	18.3	507.1	7	508.0	19
M-C106	496.5	6	497.0	20.5	516.5	6	516.5	18.7	547.9	6	547.9	19.6
M-C107	405.7	6	405.7	16.9	458.0	6	458.0	17.3	481.1	6	482.2	19.5
M-C108	583.4	6	584.3	16.6	643.7	7	643.7	18.8	866.4	8	866.4	19.4
M-C109	376.2	5	376.2	18	479.1	5	479.4	19.1	461.8	5	463.9	18.7
M-RC101	1110.8	9	1110.8	16.4	1146.6	9	1146.6	18.7	1027.1	8	1027.1	19.5
M-RC102	944.6	8	944.6	16.1	912.4	8	914.3	16.9	909.6	7	915.0	17.7
M-RC103	845.1	7	845.5	17.3	1041.3	8	1042.4	18.9	891.3	7	892.7	19.1
M-RC104	660.2	6	660.2	15.4	795.3	6	795.3	16.4	795.3	6	795.3	15.6
M-RC105	1075.6	9	1075.6	16.8	1212.8	9	1213.2	17.9	1144.4	9	1144.4	19.6
M-RC106	927.5	8	927.5	17.5	953.1	8	953.4	19.3	974.2	8	974.5	20.2
M-RC107	951.9	8	951.9	19.2	1341.3	9	1341.8	19.6	1278.3	9	1283.3	23.7
M-RC108	777.1	6	778.4	16.4	779.3	6	779.3	18.7	847.0	6	852.9	19.4
M-R201	764.1	5	764.1	17.3	784.1	5	784.9	17.4	781.0	4	781.0	17
M-R202	685.6	5	695.8	17.4	712.2	5	719.0	19.6	699.2	5	701.7	18.7
M-R203	581.4	3	581.4	17.7	584.5	3	584.5	18.8	581.4	3	581.4	18.2
M-R204	473.0	2	483.7	20.3	485.3	2	488.9	30	493.6	2	493.9	21.1
M-R205	637.6	3	643.2	18.5	635.9	4	637.0	17.7	676.9	4	681.3	17.7
M-R206	551.3	2	551.3	16.5	551.3	2	551.7	38.7	564.3	2	564.7	18.2
M-R207	525.2	3	529.7	19.1	526.8	3	530.1	23.4	545.6	2	546.1	20.1
M-R208	462.0	2	462.3	20.4	457.7	2	457.7	21.2	476.7	2	478.2	22.1
M-R209	593.4	3	593.9	15.7	591.8	3	592.9	17.4	595.5	3	595.5	17.4
M-R210	601.0	3	604.7	15.9	607.8	3	609.6	16.6	607.8	3	610.3	16.4
M-R211	524.6	2	524.9	17.2	524.4	2	524.4	18.2	542.2	2	542.3	18.8
M-C201	398.0	3	398.0	17.6	411.4	3	411.4	17.8	412.1	3	412.1	18.2
M-C202	411.5	2	411.5	19.5	388.1	2	388.1	18.4	411.5	2	411.5	19.9
M-C203	371.2	3	371.2	19.6	379.3	3	379.3	20.7	379.3	3	379.3	22.1
M-C204	362.8	2	363.2	21.3	370.0	2	370.3	21.8	365.7	2	366.1	22.1
M-C205	375.4	3	375.4	16.9	370.0	3	370.0	16.9	375.0	3	375.0	18.2
M-C206	372.8	3	372.8	15.9	383.7	3	383.7	17.1	431.7	2	431.7	17.8
M-C207	384.4	3	384.5	17.8	385.2	3	385.2	18.8	401.6	3	401.6	17.7
M-C208	412.6	2	412.6	16.7	418.0	2	418.0	19.2	418.2	2	418.2	21.8
M-RC201	732.3	5	732.3	17.5	732.3	5	732.3	18.2	735.7	5	735.7	19.5
M-RC202	681.8	4	683.6	17.7	694.3	3	694.3	20.6	716.5	4	716.5	20
M-RC203	595.8	4	598.7	18.4	668.6	3	669.1	19.4	702.6	2	704.7	18.9
M-RC204	485.3	2	485.6	20	485.3	2	485.6	20.3	485.3	2	485.3	19.6
M-RC205	746.0	4	747.4	18.6	759.1	4	759.1	19.4	784.5	4	786.2	20.3
M-RC206	686.2	4	688.9	17.5	685.7	4	685.7	18	686.2	4	686.2	20.3
M-RC207	633.3	3	633.9	15	633.3	3	633.3	16	633.3	3	633.3	16
M-RC208	486.2	3	491.1	17.4	524.2	3	533.2	17.9	558.3	2	560.9	18.2
Average	642.9	5.3	644.1	17.9	680.6	5.4	681.4	19.8	687.4	5.3	688.3	19.1

Table E.2: ALNS results for 50-node problems using Category 4, 5 and 6 parameters

The Proposed ALNS Results for Instances with 50 Nodes												
	Category 4				Category 5				Category 6			
Instances	best dist.	best #veh.	avg. dist.	avg. time (secs)	best dist.	best #veh.	avg. dist.	avg. time (secs)	best dist.	best #veh.	avg. dist.	avg. time (secs)
M-R101	985.8	11	985.8	25	1106.7	13	1109.2	29.5	1097.6	12	1099.6	30.8
M-R102	906.9	10	912.6	23.9	931.5	9	935.2	27.1	1047.7	11	1053.2	32.2
M-R103	784.2	9	784.2	21.6	768.3	8	768.3	22.9	823.3	9	823.3	24.2
M-R104	592.4	5	594.1	22.4	620.1	5	628.9	23.4	609.4	5	612.1	25.6
M-R105	793.3	8	793.3	23.5	859.1	9	859.4	28.3	866.4	9	866.4	26.7
M-R106	809.4	7	809.8	21.4	815.9	7	815.9	22.7	821.2	7	822.0	23.6
M-R107	684.6	6	684.6	24.2	719.7	6	721.2	27.4	725.3	6	727.6	25.5
M-R108	570.6	5	572.5	21.3	574.9	5	576.0	22.9	589.7	5	592.5	23.4
M-R109	854.0	8	858.3	24.6	868.5	8	869.0	22.8	906.9	8	907.5	28.1
M-R110	750.9	7	752.7	22.1	714.3	7	714.7	20.9	756.6	7	758.7	21.9
M-R111	702.3	7	703.1	25.2	759.0	7	760.2	23	777.3	8	783.2	26.6
M-R112	667.5	6	669.9	22.9	686.9	6	686.9	23.9	771.9	7	776.5	26.2
M-C101	496.1	7	496.1	22.9	513.8	7	513.8	25.5	545.7	7	545.7	27.6
M-C102	462.3	6	462.3	20.4	454.2	6	454.2	20.8	538.5	7	539.8	26
M-C103	458.6	6	458.6	24.4	541.0	6	541.0	27.7	702.5	7	712.8	34.5
M-C104	380.0	5	380.4	23.5	423.6	5	423.6	23.8	428.1	4	433.6	26.8
M-C105	454.3	6	454.3	21.6	464.7	6	470.2	23.2	559.5	7	561.6	30.2
M-C106	492.0	6	496.1	22.5	492.0	6	492.0	22.8	562.0	7	562.0	25.6
M-C107	483.5	7	486.7	26.4	500.2	7	500.2	25.3	484.0	6	489.5	28
M-C108	472.7	6	472.7	21.5	594.8	5	598.1	26.5	563.6	6	570.0	26.8
M-C109	371.1	5	371.4	20.5	380.7	5	380.7	21.6	383.2	5	383.2	23.2
M-RC101	1013.9	8	1013.9	20.4	1080.3	9	1080.3	22.2	1070.1	9	1070.1	24
M-RC102	859.1	8	859.1	20.1	984.1	9	984.1	21.7	1030.8	9	1030.8	24.6
M-RC103	806.9	7	808.7	20.5	960.1	8	960.1	26.7	996.7	8	996.7	24.7
M-RC104	648.0	6	648.0	23.5	657.8	6	658.6	23.2	779.4	6	785.6	29.3
M-RC105	861.9	7	861.9	23.2	1415.4	11	1416.1	27.6	1292.4	10	1293.2	31.7
M-RC106	702.7	6	702.7	22.6	784.6	7	784.6	21.8	838.2	7	838.3	28.5
M-RC107	841.1	7	841.9	22.4	924.1	7	925.3	23.8	908.1	7	909.3	26.7
M-RC108	625.7	6	625.7	22.6	673.5	6	673.5	26.8	826.3	7	826.4	26.9
M-R201	718.1	5	718.6	20.6	725.8	5	730.8	21.9	718.2	4	718.8	21.9
M-R202	657.2	4	659.0	21.6	658.2	4	661.1	23.4	674.7	4	676.8	24.1
M-R203	569.4	3	569.4	23.4	579.3	3	579.7	24.2	578.9	3	578.9	26.2
M-R204	450.8	2	450.8	25.8	459.2	2	461.5	26.9	468.0	2	469.9	29
M-R205	601.5	3	601.9	21.2	612.3	3	612.3	23.3	616.7	3	617.2	23.4
M-R206	522.7	2	524.4	24	572.8	2	576.9	29.3	565.0	2	568.6	28.3
M-R207	504.7	2	507.7	26.7	516.2	2	517.6	33.4	516.2	2	516.8	32.7
M-R208	429.4	2	429.9	30.2	429.4	2	429.4	31.7	439.9	2	440.2	37.6
M-R209	563.0	3	563.2	23.1	563.0	3	563.1	24.7	573.7	3	575.1	25.3
M-R210	587.1	2	590.0	23.9	587.1	2	590.2	25.5	588.6	3	590.8	26.3
M-R211	505.2	3	510.3	24.9	529.9	2	530.5	28.1	535.7	2	539.5	30.9
M-C201	358.7	3	358.7	22.8	358.7	3	358.7	24.4	344.6	3	344.6	25.3
M-C202	377.8	2	377.8	25.5	377.8	2	377.8	28.9	389.0	2	389.0	31.7
M-C203	352.1	3	352.1	24.4	360.0	3	360.0	27.8	362.6	3	362.6	27.4
M-C204	354.4	2	359.6	28.8	367.2	2	370.2	32.7	372.2	2	373.6	34.6
M-C205	381.6	3	382.1	21.6	383.5	3	387.7	22.8	381.0	3	382.4	24.3
M-C206	414.0	2	414.0	23.4	414.0	2	414.0	24.4	432.8	2	432.8	28
M-C207	362.6	3	362.6	23.6	418.0	3	418.7	26.4	458.2	3	458.6	25.3
M-C208	355.4	2	355.4	25.9	352.7	2	353.2	27.5	360.7	2	360.7	29.4
M-RC201	669.8	5	671.1	21.3	738.4	5	739.9	24.8	768.8	4	769.6	29.3
M-RC202	640.2	4	640.2	22.2	734.0	3	734.1	24.9	703.3	3	706.2	28.1
M-RC203	596.3	3	597.1	22.3	634.3	3	635.5	23.1	639.7	3	640.2	27.8
M-RC204	475.4	2	476.7	25.6	475.4	2	475.4	22.4	499.0	2	504.4	26.2
M-RC205	670.4	5	671.4	23.4	696.0	4	697.0	25.5	733.6	4	733.6	26.3
M-RC206	645.5	4	645.6	23.7	680.4	4	683.3	25.5	678.5	4	679.0	27.7
M-RC207	555.7	3	555.7	22.3	555.7	3	556.4	25.5	555.7	3	556.2	24.7
M-RC208	446.6	3	449.1	22.7	456.7	3	457.1	24.4	458.9	3	458.9	26.6
Average	594.6	5.0	595.6	23.3	634.0	5.1	635.2	25.2	655.7	5.2	657.4	27.3

Table E.3: ALNS results for 50-node problems using Category 7, 8, 9 and 10 parameters

The Proposed ALNS Results for Instances with 50 Nodes																
	Category 7				Category 8				Category 9				Category 10			
Instances	best dist.	best #veh.	avg. dist.	avg. time (secs)	best dist.	best #veh.	avg. dist.	avg. time (secs)	best dist.	best #veh.	avg. dist.	avg. time (secs)	best dist.	best #veh.	avg. dist.	avg. time (secs)
M-R101	1266.1	13	1266.1	28.1	1054.9	11	1055.1	23.1	1049.3	12	1049.3	20.6	1105.2	12	1105.7	35.4
M-R102	923.5	10	923.6	25.5	860.7	9	860.7	19.6	879.7	10	880.8	22	928.2	10	930.7	32.5
M-R103	955.3	10	955.3	23.1	777.9	9	778.0	15.9	873.3	9	873.3	17.7	716.1	7	716.1	29.7
M-R104	875.7	7	878.6	25.5	564.6	5	565.4	18.8	566.7	5	567.0	18.7	595.8	5	597.9	34.1
M-R105	1057.0	11	1058.4	25.3	841.9	9	841.9	20.5	927.9	10	928.4	19.9	854.1	9	854.1	29.1
M-R106	996.7	9	996.7	27.2	789.9	8	792.4	17.6	841.9	8	844.4	19.1	861.3	9	861.3	33
M-R107	759.8	7	759.8	24.3	685.9	6	690.2	19.4	725.2	7	728.5	18.8	659.2	6	660.8	29.4
M-R108	649.4	6	649.4	20.8	555.9	5	555.9	16.3	615.9	5	622.9	17.9	592.4	5	592.4	30.7
M-R109	968.3	9	970.9	24.3	786.0	7	787.8	16.9	893.3	9	898.3	19.7	827.9	8	834.4	39.4
M-R110	1085.1	9	1086.7	24.8	751.8	7	751.8	16.1	733.3	7	734.4	17.9	744.4	7	746.1	27
M-R111	962.1	8	962.2	24.7	673.9	6	674.4	15.9	711.8	6	713.7	17.9	733.6	7	734.1	31.5
M-R112	745.2	7	747.9	24.1	636.6	6	636.6	19.3	660.8	6	663.4	19.1	644.0	6	652.4	30.3
M-C101	618.1	8	618.1	24.8	410.2	5	422.9	18.8	422.3	6	422.3	22.9	384.4	5	384.4	28.8
M-C102	760.6	9	760.6	26.8	457.0	6	458.0	17.6	539.6	7	539.6	19	481.0	6	481.0	28.5
M-C103	550.3	6	550.3	28.3	472.4	5	473.8	19.7	473.8	5	473.8	22.1	486.8	6	487.1	33.5
M-C104	512.4	5	512.5	26.2	396.4	4	398.8	17.6	430.2	4	430.2	19.2	379.1	4	379.1	29.4
M-C105	503.6	7	503.6	23.2	459.5	6	459.7	18.2	429.9	6	429.9	18.9	563.3	7	563.6	30.3
M-C106	570.9	7	570.9	24.3	460.9	6	460.9	16.7	499.3	7	499.3	21	485.8	6	485.8	25.5
M-C107	494.7	6	497.2	25.5	445.9	6	445.9	16.9	417.4	6	417.4	18.9	526.2	7	528.7	32.5
M-C108	660.3	7	661.6	23.1	500.9	6	501.6	16.7	490.3	6	490.3	18.1	451.4	5	451.4	28.4
M-C109	495.1	5	495.4	24.6	366.5	5	366.5	15.5	383.4	5	383.4	17.7	367.7	5	368.0	30
M-RC101	1155.8	9	1155.8	23.8	942.6	8	942.6	16.4	1056.0	9	1057.9	19.3	960.8	9	960.8	30.4
M-RC102	901.6	7	901.6	21.7	792.8	7	792.8	16.2	810.4	7	810.4	16.3	1202.5	10	1210.2	33.3
M-RC103	1666.2	12	1666.2	27.6	789.6	7	789.6	17.7	821.3	7	821.9	17.1	1064.3	8	1065.2	29.8
M-RC104	871.7	6	871.7	20.9	640.6	6	640.6	16.7	672.8	6	672.8	18.5	685.9	6	685.9	31.9
M-RC105	1547.7	11	1547.7	28.1	1083.9	8	1083.9	19.2	1088.6	8	1088.6	19.4	864.1	7	864.1	33.7
M-RC106	995.3	8	996.1	23.4	800.4	7	802.2	16.9	936.6	7	937.3	18.8	1034.4	8	1034.4	30.9
M-RC107	1648.6	11	1648.7	27.1	832.4	7	832.7	18.7	881.1	7	881.1	19.4	865.3	7	865.8	30
M-RC108	839.5	6	842.3	24	655.5	6	655.5	18.7	702.8	6	702.8	19.4	697.7	6	697.7	31.6
M-R201	827.8	5	838.2	22	707.6	4	713.0	16.7	728.4	4	728.4	19.2	682.3	4	683.7	25
M-R202	744.7	4	747.9	25.2	652.5	5	653.8	18.2	679.5	4	680.6	19	636.0	3	636.0	32.6
M-R203	605.6	2	609.1	25.1	577.5	3	577.5	17.6	564.3	3	564.3	20.2	553.6	3	553.6	32.7
M-R204	493.3	2	493.3	26.6	447.6	2	447.9	20.3	463.2	2	463.2	23.6	448.8	2	457.6	33.3
M-R205	647.8	3	647.9	22.7	599.7	3	600.9	18.5	621.5	3	621.9	18	617.1	3	618.3	30.9
M-R206	551.3	2	551.5	26.1	556.0	2	557.0	20.2	571.8	3	573.3	21.2	530.2	2	532.9	35.8
M-R207	526.8	3	530.0	28.6	512.4	2	517.2	21.1	528.8	2	533.0	23	469.0	2	469.5	32.8
M-R208	468.1	2	468.4	27.7	429.4	2	429.4	24	439.8	1	441.8	26.9	414.0	2	414.7	40.8
M-R209	598.0	3	598.7	22.6	562.3	3	563.0	19	581.9	3	582.7	19.6	551.4	3	552.1	31.6
M-R210	612.3	2	621.3	23.3	568.9	4	572.0	16.5	591.6	3	595.4	17.5	566.6	2	570.7	36.5
M-R211	540.4	2	542.2	26.7	505.1	2	507.0	19.5	505.6	2	505.6	20	506.2	2	509.6	34.7
M-C201	413.0	3	413.0	24.3	358.3	3	358.3	18.4	381.9	3	381.9	19.3	379.6	3	379.6	30
M-C202	413.2	2	413.2	27	363.9	3	363.9	20.3	366.1	3	366.1	24	396.9	2	399.5	37
M-C203	388.3	3	388.3	26.1	349.5	3	349.5	18.5	368.0	3	368.0	20.9	390.9	3	391.3	35.9
M-C204	374.1	2	374.1	30.2	357.5	2	361.6	22.5	341.4	2	341.4	22.8	346.2	2	350.3	32.7
M-C205	372.4	3	372.4	21.6	356.2	3	356.2	17.5	362.9	3	362.9	18.3	336.5	2	336.5	31.2
M-C206	387.1	2	387.1	24.7	414.0	2	414.0	16.8	427.7	2	427.7	19.5	376.0	2	376.0	35.2
M-C207	422.0	3	422.4	25.8	375.6	3	375.6	18.8	379.3	3	379.3	20.3	367.2	3	367.2	28.4
M-C208	418.0	2	418.1	26.2	333.5	2	333.5	17.5	342.5	2	342.5	20.7	359.9	2	359.9	33.9
M-RC201	746.7	5	746.7	22.1	700.6	5	700.8	17.2	693.3	5	693.3	17.6	667.2	5	667.3	34.7
M-RC202	700.4	4	700.4	25.4	714.9	4	716.0	21.2	708.2	4	709.7	19.5	642.7	4	643.3	30.3
M-RC203	722.0	2	734.4	24	598.3	3	599.0	18.4	595.2	3	596.0	20.1	591.4	3	596.0	32.6
M-RC204	517.8	2	528.6	26.9	480.1	2	480.3	18.9	472.4	2	472.4	21.2	459.3	2	459.3	37.4
M-RC205	770.6	4	772.1	26.2	661.8	4	661.8	21.7	702.2	3	702.2	23.5	680.0	4	681.2	35.4
M-RC206	686.2	4	686.2	24.2	667.4	3	670.7	18.7	681.6	4	686.2	19.7	607.5	4	607.5	33.8
M-RC207	644.9	3	647.1	22.8	548.6	3	548.6	19.2	551.8	3	551.8	19.4	552.4	3	552.4	27.3
M-RC208	534.8	3	537.7	24.4	458.9	3	458.9	19.3	453.4	3	463.4	20.1	463.6	2	464.0	33.2
Average	735.1	6	736.5	25.0	595.5	5	596.5	18.5	618.6	5	619.6	19.8	613.5	5	614.8	32.1

Appendix F: Detailed ALNS results of new benchmark instances with 100 nodes

We report here the detailed ALNS results of new benchmark instances with 100 nodes.

Table F.1: ALNS results for 100-node problems using Category 1, 2 and 3 parameters

The Proposed ALNS Results for Instances with 100 Nodes												
	Category 1				Category 2				Category 3			
Instances	best dist.	best #veh.	avg. dist.	avg. time (secs)	best dist.	best #veh.	avg. dist.	avg. time (secs)	best dist.	best #veh.	avg. dist.	avg. time (secs)
M-R101	1723.9	20	1727.1	91.1	1721.6	19	1739.7	94.9	1678.4	19	1679.8	88.9
M-R102	1477.8	16	1480.4	81.4	1449.7	16	1456.0	83.4	1449.1	15	1453.3	85.8
M-R103	1251.9	14	1254.3	80.7	1277.6	14	1278.4	86	1263.0	14	1266.6	83.5
M-R104	1023.4	10	1030.8	80.1	1027.8	10	1037.2	84.2	1064.3	10	1070.5	82.3
M-R105	1397.3	15	1402.8	76.2	1537.6	16	1543.0	83.8	1505.6	16	1511.2	85.7
M-R106	1278.6	13	1280.4	77.9	1295.6	13	1300.7	83.1	1332.3	13	1341.5	83
M-R107	1098.2	12	1100.4	76.8	1098.7	12	1100.6	81.7	1095.4	12	1097.5	80.4
M-R108	968.6	9	975.6	69.5	1017.0	10	1017.7	71.3	1171.3	11	1183.4	79.5
M-R109	1236.9	13	1244.9	80	1238.9	13	1252.8	77.6	1284.2	13	1301.6	81.7
M-R110	1074.7	11	1081.0	71.1	1100.6	12	1109.7	78.6	1125.4	11	1133.1	79.3
M-R111	1037.7	11	1055.5	77.1	1207.0	12	1213.4	81.7	1100.0	12	1107.4	86.5
M-R112	927.8	10	933.1	69.2	929.7	10	936.8	72.9	980.3	10	985.5	80.2
M-C101	1220.8	14	1225.7	81	1211.7	13	1211.7	82.7	1292.6	14	1293.5	89.5
M-C102	1006.7	11	1006.7	76.9	1009.8	11	1009.8	84.6	1091.9	11	1092.8	86.6
M-C103	957.4	10	958.5	81.9	1058.9	10	1062.9	90.5	1017.1	10	1017.1	91
M-C104	894.3	9	900.2	83.9	929.6	9	932.8	89.3	924.4	9	929.7	80.5
M-C105	1040.9	12	1040.9	75.9	1104.9	12	1104.9	77.7	1217.0	14	1227.2	88.8
M-C106	1064.6	13	1064.6	80.2	1148.0	14	1148.3	89.6	1145.0	13	1145.4	88.5
M-C107	1199.7	12	1220.9	80.6	1269.7	12	1272.1	86.6	1271.8	14	1271.8	90.4
M-C108	1007.5	11	1010.3	78.3	1087.5	11	1094.2	84.4	1047.1	10	1055.7	79.5
M-C109	847.4	10	847.4	78.4	864.0	10	865.3	80.3	1018.3	11	1030.4	86.6
M-RC101	1712.4	16	1713.9	78.8	1791.3	17	1792.8	84.9	1782.5	17	1786.6	79.1
M-RC102	1619.4	15	1624.0	85.7	1534.0	14	1534.6	84.4	1617.2	15	1619.2	85.7
M-RC103	1285.1	12	1290.7	78.8	1347.9	11	1360.1	86	1435.9	12	1444.1	87.1
M-RC104	1127.2	10	1135.2	75.3	1146.0	10	1148.9	79.2	1164.0	10	1179.1	86.6
M-RC105	1391.7	14	1393.0	74.2	1389.1	14	1390.5	78.4	1390.2	14	1390.6	78
M-RC106	1380.5	13	1392.9	75.6	1493.4	14	1498.8	85	1603.7	14	1616.2	93.8
M-RC107	1291.5	12	1293.9	79.2	1404.1	12	1411.3	85.1	1487.0	13	1493.0	80.7
M-RC108	1169.8	11	1182.9	78.9	1202.4	11	1205.9	80.1	1334.7	11	1344.1	80.5
M-R201	1057.9	6	1069.7	73.6	1074.4	6	1080.8	76.8	1088.9	5	1094.8	81.3
M-R202	1005.1	6	1011.2	69.8	1035.6	6	1044.7	77.2	1057.2	6	1069.6	80.9
M-R203	833.9	6	842.8	85.6	842.5	6	853.8	93.2	841.4	5	848.8	87.4
M-R204	716.8	4	720.3	76.9	716.0	3	721.4	83.8	719.5	3	729.8	88
M-R205	901.1	5	914.3	69.9	915.0	6	924.0	75.1	936.2	5	950.0	79.8
M-R206	831.7	4	837.7	81.1	849.7	4	851.5	88.8	851.1	4	855.9	96.6
M-R207	774.8	4	788.1	78.4	780.6	3	786.0	81.8	793.8	3	796.0	83.9
M-R208	668.5	3	673.9	84.2	679.1	3	683.8	89.6	680.3	3	689.9	96.9
M-R209	841.9	4	850.8	72.7	847.1	5	859.3	74.9	847.5	5	857.3	79.6
M-R210	870.7	6	878.8	76	888.6	5	899.7	81.1	889.2	5	898.4	86.2
M-R211	739.1	3	746.3	78.1	738.8	4	746.1	86	729.3	4	748.7	83
M-C201	638.3	4	638.3	73.1	638.3	4	638.3	77.1	641.9	4	641.9	79.7
M-C202	631.2	4	631.2	77.5	633.1	4	633.1	78.9	667.7	4	667.7	83.4
M-C203	570.2	3	570.2	75.4	639.1	3	639.3	84.8	625.5	3	625.8	85.8
M-C204	601.1	3	601.1	82.9	632.6	3	632.6	86.9	632.6	3	632.6	86
M-C205	645.8	3	645.8	72.7	645.8	3	645.8	80.2	686.4	4	686.4	77.5
M-C206	599.8	3	599.8	73.5	602.8	3	602.8	83	616.4	3	616.4	84.3
M-C207	615.8	3	615.8	73.9	617.4	3	617.4	81.2	617.4	3	617.6	82.1
M-C208	646.3	3	651.2	78.8	646.2	3	647.4	84.8	660.8	3	663.5	89.7
M-RC201	1250.4	7	1252.1	76	1240.4	6	1257.2	80	1258.9	7	1262.9	84
M-RC202	1091.3	6	1097.2	76.3	1058.8	7	1063.6	81.3	1108.3	6	1113.1	87.7
M-RC203	905.7	4	909.6	79.2	904.1	4	911.1	84.6	925.1	4	926.9	93.3
M-RC204	750.5	3	753.5	81.5	801.5	3	806.5	87.5	747.1	3	758.1	86.8
M-RC205	1158.1	7	1160.2	78.9	1148.7	8	1155.0	88.1	1164.0	8	1182.2	84.6
M-RC206	1019.8	5	1028.4	70.1	1048.6	5	1062.3	79	1064.8	4	1079.4	82.4
M-RC207	970.9	5	986.4	70.6	998.7	6	1010.8	83.2	1014.6	5	1039.6	79.4
M-RC208	793.9	4	800.3	78.9	807.5	4	816.2	86.3	819.0	4	825.5	87.4
Average	1015.1	8.5	1020.4	77.5	1041.5	8.6	1046.8	82.9	1063.8	8.6	1070.5	84.8

Table F.2: Results for 100-node problems using Category 4, 5 and 6 parameters

The Proposed ALNS Results for Instances with 100 Nodes												
	Category 4				Category 5				Category 6			
Instances	best dist.	best #veh.	avg. dist.	avg. time (secs)	best dist.	best #veh.	avg. dist.	avg. time (secs)	best dist.	best #veh.	avg. dist.	avg. time (secs)
M-R101	1572.1	19	1574.6	100.9	1677.6	19	1683.9	107.6	1641.0	19	1645.5	109.9
M-R102	1289.4	14	1292.3	103.6	1328.3	16	1330.0	109.5	1322.0	15	1324.2	119.6
M-R103	1105.7	12	1110.7	104.2	1125.3	13	1128.2	109.5	1157.5	14	1159.4	114.4
M-R104	888.5	9	897.3	90.9	912.8	9	917.7	101	919.9	9	926.0	100.3
M-R105	1418.9	15	1425.1	96.7	1495.5	16	1498.9	105.2	1496.6	15	1498.8	108.6
M-R106	1116.7	12	1128.7	104.5	1116.4	12	1116.8	100.8	1182.1	13	1190.4	121.6
M-R107	1003.9	11	1009.8	94.1	1014.2	11	1020.4	102.6	1026.1	11	1030.6	107.1
M-R108	866.4	9	868.1	97.4	896.0	9	904.4	101.1	895.4	9	903.0	127.2
M-R109	1090.2	11	1099.6	106.7	1101.2	11	1111.1	108.7	1128.1	12	1131.5	144.8
M-R110	1062.7	11	1069.4	94.5	1060.4	10	1070.1	97.4	1093.7	10	1102.6	107.4
M-R111	964.6	10	971.3	95.6	978.8	10	981.8	92.5	994.4	11	1001.3	101.3
M-R112	858.7	9	871.2	90.2	899.7	9	909.3	96.4	936.6	9	949.4	105.5
M-C101	1113.1	13	1113.5	99.9	1167.5	13	1167.5	115.7	1261.9	14	1277.9	114.1
M-C102	999.5	12	1003.7	95.1	988.8	11	988.9	97.2	1122.1	12	1122.4	109.6
M-C103	937.2	10	942.7	98	1023.9	10	1025.5	110.2	1027.5	10	1030.0	100.6
M-C104	821.2	8	827.9	100.5	823.5	8	835.4	108.6	899.2	9	904.8	121.9
M-C105	1164.9	12	1179.1	102	1205.0	13	1209.2	106.9	1266.4	13	1269.2	122.2
M-C106	1097.5	12	1099.6	98.6	1157.0	13	1164.7	110.9	1230.0	13	1234.7	119.6
M-C107	1082.0	12	1083.6	97.8	1083.3	12	1090.7	101.8	1168.6	13	1170.1	121.2
M-C108	930.7	10	945.0	97.8	952.7	10	954.8	105.6	933.8	11	933.8	110.2
M-C109	850.0	10	855.4	108	885.9	10	888.3	117.3	899.4	10	906.4	115.5
M-RC101	1476.2	14	1480.0	95.6	1686.0	16	1690.8	105.9	1835.1	17	1867.7	119.2
M-RC102	1327.7	12	1332.3	95.6	1346.8	12	1350.0	101.8	1336.8	13	1339.4	101.6
M-RC103	1281.0	11	1305.1	102.4	1330.5	12	1342.1	109.7	1341.7	12	1348.0	118
M-RC104	996.5	9	997.5	93.1	1001.1	9	1001.8	93.8	1011.8	9	1013.1	98.8
M-RC105	1287.9	13	1291.3	96.2	1281.7	13	1284.8	99.2	1323.6	13	1336.3	118.2
M-RC106	1278.4	11	1293.7	106.5	1294.7	12	1311.9	109.4	1395.0	12	1409.6	118.4
M-RC107	1181.3	11	1182.0	96.8	1165.1	11	1167.4	96.4	1184.3	11	1186.1	100.3
M-RC108	1044.1	10	1053.5	107.3	1083.0	10	1089.8	108.5	1185.7	10	1192.4	117.5
M-R201	1013.6	6	1022.2	86.2	1015.6	6	1022.3	90.8	1022.2	6	1030.4	96.1
M-R202	993.5	5	1002.1	99	998.2	5	1006.6	110.3	1001.7	5	1008.2	121.4
M-R203	789.3	5	793.7	98.4	796.8	5	805.9	110.7	794.9	6	814.9	106.9
M-R204	642.0	3	645.5	105.7	646.7	3	650.4	116.3	650.9	3	656.6	116.3
M-R205	878.3	5	884.8	96.5	884.5	4	892.7	104.3	884.3	4	890.0	107.2
M-R206	795.1	4	803.8	101.8	804.2	4	809.0	114.2	812.1	4	815.2	109.8
M-R207	760.9	4	766.7	102.6	768.5	4	774.2	119.9	765.5	4	771.6	111.3
M-R208	624.0	3	628.3	121	625.0	3	628.3	136.3	627.0	3	630.4	143.6
M-R209	780.4	4	791.4	98.8	781.5	4	793.4	111	785.9	4	796.8	108.6
M-R210	821.4	4	827.8	98.5	825.0	4	834.3	105.3	826.3	5	839.0	117.7
M-R211	662.1	3	664.7	113	678.2	4	683.5	111.5	676.8	4	683.4	111
M-C201	639.0	4	639.0	96.2	647.1	4	647.1	103.8	647.1	4	647.1	109.1
M-C202	602.0	3	602.0	102.7	620.7	3	622.3	108.4	658.5	3	667.0	118.1
M-C203	591.6	3	591.6	110.3	591.6	3	591.6	118.9	608.4	3	613.6	128.2
M-C204	603.4	3	604.5	110	604.0	3	606.2	121.1	603.6	3	604.6	120.2
M-C205	626.2	3	631.1	92.7	618.3	3	618.4	98.2	643.7	3	649.2	104.3
M-C206	566.4	3	566.4	89.6	570.3	3	570.3	93.5	605.0	3	615.3	117.4
M-C207	608.7	3	610.7	101.8	575.8	3	576.3	105.7	647.4	3	648.2	127.2
M-C208	602.2	3	604.6	109.4	618.1	3	620.7	116.2	626.6	3	626.8	111.3
M-RC201	1117.5	6	1126.1	94	1150.8	6	1159.0	97.5	1176.2	6	1183.0	105.4
M-RC202	994.7	6	998.0	102.1	987.3	6	1002.0	108.1	1006.5	6	1022.7	125.6
M-RC203	810.7	4	818.8	99.4	844.4	4	850.6	108.4	873.5	4	876.3	118.9
M-RC204	728.5	3	734.6	104.4	739.8	3	744.0	116.8	744.7	3	757.7	129.4
M-RC205	1072.4	7	1080.4	94.4	1156.3	7	1162.3	108.8	1164.8	6	1176.7	122.4
M-RC206	1044.7	5	1055.9	91.3	1062.1	5	1075.0	100.9	1030.0	4	1040.9	106.8
M-RC207	901.8	6	916.6	94.9	911.4	5	920.1	104.8	918.8	5	922.8	104.8
M-RC208	717.5	4	737.1	100.9	739.9	4	749.2	109.7	732.2	4	748.9	116.8
Average	948.1	7.9	954.5	99.8	970.4	8.1	975.9	106.8	995.6	8.2	1002.5	114.5

Table F.3: Results for 100-node problems using Category 7, 8, 9 and 10 parameters

The Proposed ALNS Results for Instances with 100 Nodes																
	Category 7				Category 8				Category 9				Category 10			
Instances	best dist.	best #veh.	avg. dist.	avg. time (secs)	best dist.	best #veh.	avg. dist.	avg. time (secs)	best dist.	best #veh.	avg. dist.	avg. time (secs)	best dist.	best #veh.	avg. dist.	avg. time (secs)
M-R101	1719.3	19	1719.6	120.1	1558.0	18	1559.9	87.2	1593.6	20	1598.4	90.8	1586.9	18	1589.7	133.6
M-R102	1481.2	16	1485.3	110.1	1277.3	15	1279.7	80.9	1353.9	14	1358.8	89.6	1270.1	14	1279.2	128.9
M-R103	1306.9	14	1309.4	110.1	1086.3	13	1086.4	78.4	1100.7	13	1102.0	84.8	1023.7	12	1024.5	134.2
M-R104	1057.7	10	1072.3	115.6	893.7	9	901.6	74	931.5	9	940.0	83.8	917.8	9	926.9	142.5
M-R105	1531.5	16	1543.6	114.4	1349.7	15	1362.9	76.8	1304.9	13	1316.7	87.4	1244.0	13	1252.3	138
M-R106	1361.1	13	1367.7	112.4	1099.1	12	1100.8	76.5	1205.4	13	1220.8	89.7	1094.2	11	1106.9	133
M-R107	1129.3	12	1135.3	110.7	970.2	10	973.5	78.1	1081.5	11	1086.0	87	998.5	10	1005.3	145.1
M-R108	1053.2	10	1068.3	96.3	838.2	9	841.4	75	885.8	9	892.6	85.7	824.3	8	829.5	132.9
M-R109	1287.7	13	1292.2	100	1070.5	11	1071.9	74.6	1110.5	12	1119.7	87.2	1030.8	10	1039.5	137.2
M-R110	1134.5	11	1143.5	103.1	1033.1	10	1038.0	73	1086.2	11	1092.8	84.2	1025.4	10	1033.5	137.6
M-R111	1387.8	12	1392.5	106.9	964.6	10	971.0	73	1025.7	11	1030.8	85.3	894.4	10	903.1	125.1
M-R112	968.4	10	974.9	95.5	866.3	9	869.7	73.1	900.4	9	915.3	80.3	818.0	8	824.0	119.1
M-C101	1404.9	16	1405.1	105.2	1059.9	12	1066.4	86.3	1194.5	13	1194.5	86.9	1166.0	13	1171.6	135.2
M-C102	1127.6	12	1130.1	102.8	997.3	11	997.5	74.5	1069.1	11	1069.2	83.2	1141.4	13	1166.6	135.8
M-C103	1091.0	10	1094.3	110.7	958.1	10	959.9	80.4	964.0	10	964.4	90.4	958.4	10	959.4	145.7
M-C104	969.0	9	972.6	114.2	810.5	8	817.7	81.4	845.4	9	846.8	85.4	778.4	8	779.9	138.8
M-C105	1225.2	13	1225.8	101.4	1002.8	11	1005.0	72.5	1076.5	12	1080.5	84.5	1039.8	12	1042.7	130.5
M-C106	1211.4	14	1211.4	114.2	1063.9	13	1068.0	88.9	1101.6	14	1103.7	88.4	1068.1	12	1075.3	136.6
M-C107	1296.8	13	1301.8	102.8	959.9	12	959.9	77.7	1107.0	14	1107.5	89.9	1061.9	12	1062.3	126.2
M-C108	1141.8	12	1146.3	107.8	933.8	10	935.1	78.5	953.8	10	957.6	82.7	904.8	10	908.1	137.8
M-C109	892.5	10	892.5	104.5	836.4	10	836.7	79.2	885.5	9	896.7	89.1	896.4	10	902.0	144.9
M-RC101	1943.6	18	1950.4	108.5	1517.7	14	1522.4	76.4	1710.2	15	1721.4	87.7	1388.5	13	1417.2	130.5
M-RC102	1704.2	15	1707.0	115.9	1298.7	12	1302.7	75.2	1326.4	12	1328.5	84.8	1203.0	11	1217.6	127.9
M-RC103	1417.8	12	1425.7	107.7	1252.4	11	1264.0	73.2	1282.1	11	1287.6	80.9	1211.5	11	1221.0	130
M-RC104	1184.9	10	1189.3	101.8	980.7	9	983.7	72.1	1021.8	10	1022.2	85	1012.3	9	1016.6	127.7
M-RC105	1395.6	14	1396.2	104.3	1274.5	13	1275.9	77.5	1348.0	14	1350.7	90.4	1260.1	12	1260.9	130.4
M-RC106	1505.4	14	1516.0	112	1274.1	11	1279.4	83.3	1375.8	12	1395.4	91.4	1318.3	12	1323.5	143.9
M-RC107	1438.4	13	1454.2	109.2	1153.2	10	1155.1	73.4	1192.8	11	1196.0	83.3	1167.8	11	1176.6	181.6
M-RC108	1246.9	11	1269.0	103.7	1004.0	10	1006.7	74.8	1128.8	10	1140.2	86.7	987.3	9	999.1	127.8
M-R201	1077.5	6	1081.9	98.9	1000.2	6	1010.3	67.2	1043.6	6	1052.6	79.6	997.5	5	1003.9	128.3
M-R202	1046.9	6	1063.9	108	983.5	6	988.1	79	989.8	5	996.7	88	976.0	5	979.9	139.9
M-R203	866.2	5	875.1	134.9	778.0	6	798.6	80	813.4	6	821.4	94.2	778.9	4	791.0	145
M-R204	727.1	3	734.6	115.6	644.3	3	647.4	82.3	690.4	3	695.9	85.8	622.0	3	626.2	152.4
M-R205	947.7	5	957.9	104	861.5	5	863.2	78.6	915.4	4	923.3	92.9	862.7	4	867.0	119.6
M-R206	850.3	4	851.8	113.6	803.2	4	806.3	82.5	819.1	4	823.7	99.9	773.3	4	786.0	147.6
M-R207	794.6	4	801.7	106.9	751.4	4	755.3	81.8	784.3	4	793.3	87.9	734.7	4	746.8	160.7
M-R208	676.9	3	686.9	121.8	621.1	3	624.5	92.5	645.3	3	650.4	93.4	594.4	3	598.7	162.5
M-R209	863.1	4	879.5	99.2	776.1	4	787.9	80.8	791.5	4	797.2	89	778.6	4	780.6	146.5
M-R210	906.7	5	916.4	107.6	835.9	5	839.3	74.3	837.3	4	842.4	85	825.6	4	836.8	153.7
M-R211	733.4	3	743.4	116.4	666.4	3	670.7	80	699.6	3	709.4	87	642.6	3	648.9	145.1
M-C201	673.9	5	673.9	109	598.1	3	598.1	77	661.9	5	661.9	85.4	660.3	5	660.3	148.6
M-C202	643.5	4	643.5	111.8	615.5	3	615.6	81.8	654.1	4	654.1	91.9	606.4	3	607.7	141.8
M-C203	647.7	3	648.6	117.8	594.1	3	594.1	83.3	622.2	3	633.4	91.2	552.2	3	552.3	144.8
M-C204	645.6	3	647.1	112.9	581.9	3	582.3	83.6	580.7	3	581.1	98.6	562.5	3	564.3	155.2
M-C205	645.8	3	645.8	105.2	626.2	3	627.6	72.6	632.5	3	634.0	78.6	586.6	3	586.9	149.5
M-C206	664.4	3	664.8	109.6	545.7	3	545.7	68.4	566.8	3	566.8	85.3	572.5	3	572.9	130.1
M-C207	647.0	3	647.6	117.3	565.2	3	565.2	71.3	645.5	3	646.5	82.9	593.8	3	593.8	164.4
M-C208	643.8	3	643.8	113.8	603.4	3	603.9	80.3	615.3	3	615.6	90.5	606.8	3	608.4	156.5
M-RC201	1265.4	6	1281.1	101.4	1114.2	6	1120.1	74.4	1219.6	7	1225.5	86.1	1100.7	6	1111.3	213.7
M-RC202	1089.2	6	1095.2	107.6	970.3	6	973.6	81.6	1052.3	6	1057.9	93.9	994.7	5	1008.3	205.5
M-RC203	919.3	4	922.1	111.4	811.7	4	814.2	76.4	836.6	5	842.8	86.1	837.1	4	848.8	161.2
M-RC204	753.6	3	764.6	114.5	724.2	3	731.0	81.7	741.5	3	749.7	93.1	720.6	3	722.1	241.3
M-RC205	1178.7	7	1182.7	116.7	1062.8	8	1067.4	79.6	1111.7	7	1116.5	83.1	1062.9	8	1070.6	168.5
M-RC206	1047.8	5	1055.9	101.2	1014.5	6	1026.6	71	1022.4	5	1026.8	78.6	957.0	5	982.7	146.6
M-RC207	1004.5	5	1036.8	115.1	904.8	5	911.7	76.9	964.4	6	974.2	87	926.2	5	937.3	153.6
M-RC208	795.7	4	808.4	110	721.7	4	731.2	77.4	779.9	4	787.1	94.1	704.4	4	716.9	150.7
Average	1078.1	9	1084.8	109.4	931.4	8	935.6	77.9	980.4	8	986.0	87.4	927.2	8	934.4	145.9