The Vehicle Routing Problem in Urban Logistics: Route Balancing with Time Windows

by

Banu Ulusoy

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Industrial Engineering

KOÇ UNIVERSITY

October, 2019

The Vehicle Routing Problem in Urban Logistics: Route Balancing with Time Windows

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Banu Ulusoy

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

Prof. Metin Türkay (Advisor)

Assist. Prof. Barış Yıldız

Assist. Prof. Utku Koç

Date:

To my dearest family

ABSTRACT

The vehicle routing problem (VRP) has been widely studied in operations research literature with a variety of extensions to reflect the particular conditions pertinent to real-life applications. An important characteristics of vehicle routing problems in urban areas is the time limits when the deliveries can be made. Also, there are usually a large number of vehicles serving customers in different locations in an urban environment. A routing plan should also consider balancing the workload of each vehicle in terms of total distance travelled or total time spent on the route in order to maximize employee satisfaction. In this thesis, we study the VRP with time windows and route balance. We present a mixed-integer linear programming model with the objective of minimizing the total number of routes, total distance traveled, and total time spent while providing a balance between the routes. It is possible to use exact algorithms for solving small scale problems. However, these algorithms fail to provide satisfactory solutions in a reasonable time for medium and large scale problems. We develop a scatter search-based heuristic algorithm for solving medium to large scale problems. We also add a split delivery extension, that reflects the characteristics of urban logistics, to the problem and modify our models accordingly. The models and the solution algorithm are tested on Solomon benchmark problems and real data collected from three different companies based in Turkey. The heuristic algorithm decreases the total number of routes and total time in the solutions by 15-20%. We show that the algorithm gives successful results in a short computing time and can be applicable in various areas of logistics. We also address future research areas considering environmental and social aspects of the problem.

ÖZETÇE

Araç rotalama problemi, literatürde yaygın olarak çalışılan ve geniş kapsamlı uygulamaları olan bir problemdir. Kentsel bölgelerde araç rotalama probleminin önemli bir özelliği, teslimatların yapılabileceği zaman sınırlarıdır. Şehirlerde farklı konumlardaki müşterilere hizmet veren çok sayıda araç vardır. Planlanan rotalar, çalışan memnuniyetini en üst seviyeye çıkarmak için her bir aracın iş yükünü, seyahat edilen toplam mesafeyi veya rota üzerinde harcanan toplam süreyi dengelemeyi dikkate almalıdır. Bu tezde, araç rotalama problemi zaman pencereleri ve rota dengesi ile incelenmektedir. Araçlar arasında bir denge sağlarken, toplam güzergah sayısını, toplam katedilen mesafeyi ve toplam harcanan zamanı en aza indirmek için karışık tamsayılı doğrusal programlama modeli geliştirilmiştir. Küçük boyutlu problemleri çözmek için matematiksel modeller kullanmak mümkündür, ancak bu modeller orta ve büyük ölçekli problemler için makul bir sürede tatmin edici çözümler sağlayamamaktadır. Orta ve büyük ölçekli problemleri çözmek için dağınık arama tabanlı bir sezgisel algoritma geliştirilmiştir. Şehir içi lojistiğin özelliklerini yansıtan bölünmüş teslimat uzantısı da probleme eklenmiş ve modeller buna göre modifiye edilmiştir. Matematiksel ve sezgisel algoritmalar, Solomon kıyaslama problemleri ve Türkiye'deki üç farklı şirketten toplanan gerçek veriler üzerinde test edilmiştir. Sezgisel algoritma sonucu toplam rota sayısı ve toplam süre %15-20 oranında azalmaktadır. Geliştirdiğimiz sezgisel yaklaşımın çok kısa bir hesaplama süresi içinde başarılı sonuçlar verdiği ve lojistik alanında uygulamalarının olabileceği görülmüştür. Son olarak, gelecekteki araştırmalar için çevresel ve sosyal yaklaşımlar ışığında tavsiyelerde bulunulmuştur.

ACKNOWLEDGMENTS

I would like to thank my supervisor Prof. Metin Türkay who guided me and provide invaluable support during my research. I am very fortunate to have worked with him. I also would like to thank my jury members Assist. Prof. Barış Yıldız and Assist. Prof. Utku Koç for their valuable comments and support on my thesis.

I would like to thank the members of SystemsLab whom I have worked with for three years. During my time in Koç University, I made valuable friendships. I want to thank Çiğdem Ak, Feray Tunçalp and Kiran Anwar for their support and friendship.

I would like to thank Onur Dereli in particular. I feel very fortunate to have met him at this time of my life. He has always supported me and helped me to overcome the difficulties. Once again, thank you for always being here and for your endless support.

Finally but most importantly, I would like to thank my family. I would have never achieved a little without their endless love and support. I would like to thank my father Sefa for his support and belief in me throughout my life. I am very thankful to my mother Emine for always being there for me unconditionally and for showing me how strong a woman can be in life. Finally, I would like to thank my brother Erdem. He is not just an elder brother to me, but also a very special friend and mentor. He always supports and encourages me in every possible way. I am very grateful to have a brother like him. He dedicated his Ph.D. thesis to me. Now, I want to dedicate this thesis to my dearest family, especially to my beloved brother.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

CVRP	Capacitated Vehicle Routing Problem
FMCG	Fast Moving Consumer Good
MIP	Mixed Integer Programming
NN	Nearest Neighbour
SDVRP	Split Delivery in Vehicle Routing Problem
SDVRPTWRB	Split Delivery in Vehicle Routing Problem with Time Windows and Route Balance
SS	Scatter Search
VRP	Vehicle Routing Problem
VRPRB	Vehicle Routing Problem with Route Balance
VRPTW	Vehicle Routing Problem with Time Windows
VRPTWRB	Vehicle Routing Problem with Time Windows and Route Balance
VRPTWSD	Vehicle Routing Problem with Time Windows and Split Delivery
TSP	Traveling Salesman Problem

Chapter 1

INTRODUCTION

The Vehicle Routing Problem (VRP) is one of the most studied combinatorial optimization problems in operations research and logistics. It has both theoretical insights and real-world applications. Designing effective and efficient vehicle routes does not only have an economic impact, but also affects the environmental and social aspects; organizations can reduce their transportation costs, carbon footprint and improve the working conditions of the employees.

The objective of the classical VRP is to find a set of routes which start and end at a depot, while satisfying customer demands using a set of vehicles. Each node is visited only once by a vehicle. Typical objectives of the problem are: minimization of total distance, total transportation cost, total number of vehicles, travel time balance, vehicle load balance, or weighted combination of these objectives [Toth and Vigo, 2002b].

One of the mostly studied extensions of the problem is, the capacitated vehicle routing problem (CVRP) in which vehicles have limited capacities. The distance-constrained vehicle routing problem aims to solve the problem without violating a given distance limit for each vehicle. Vehicle Routing with Time Windows (VRPTW) is another variant of the problem in which the customers have earliest and/or latest acceptance times; the service of a customer must begin within these time windows. The service start times are also defined as decision variables [Toth and Vigo, 2002a]. The vehicle routing problem with time windows and split deliveries (VRPTWSD) is a variation of the VRPTW in which more than one vehicle can serve each customer [Archetti and Speranza, 2012].

VRP and its variations, which are stated above, are NP-hard problems [Kumar and Panneerselvam, 2012]; the problem complexity increases exponentially as the problem size increases. Thus, depending on the data, the exact algorithms may only work for a small size of instances. However, well-performing heuristic algorithms can generate satisfying results. In the literature review section, we examine these heuristic algorithms.

Urban logistics is important for urban economies and requires optimal and effective design. Companies should consider traffic congestion and road- vehicle availability to minimize total length of the routes and total number of vehicles used. Organizations are constantly under pressure to consider environmental and social impacts of their operations in addition to pure economic perspective [Vega-Mejía et al., 2019]. Economic aspects can be viewed as transportation costs, fixed costs, and rental costs of vehicles. Carbon dioxide emissions of vehicles on each route can be measured to evaluate the environmental aspect. The workload equity among the drivers can evaluate the social aspect of route balancing.

This thesis studies route balancing in VRPTW in urban logistics. We consider balancing the return times of vehicles to the depot after performing the routes. The objective of the problem is to minimize the total number of routes, total distance, and total travel time while balancing the routes. The literature has various studies on route balancing; however, most of them consider balanced loads or balanced total length of the routes. We extend this view to balancing return times of the vehicles to the depot; therefore, every vehicle will start routing and depart the depot at the same time and arrive at the end of the shift at nearly the same time.

We address two problems in this thesis: route balance in vehicle routing problems with time windows (VRPTWRB), and split delivery extension of this problem (SDVRPTWRB). We develop two mixed integer programming (MIP) models and heuristic algorithms to solve them. However, recall that VRP and its extensions are NP-hard problems and can solve a small size instances. Conversely, the heuristic algorithms solve larger problems in a very short amount of time and provides satisfactory results. Our heuristic algorithms are based on the scatter search algorithm which is already used to solve the VRP models [Glover, 1977]. We performed our models on Solomon benchmark problems and three different data sets from different sectors based in Turkey: shipping data of a spare part warehouse of a large manufacturing company, warehouse shipping data of a fast-moving consumer goods (FMCG) company and shipping data of a logistics company.

The thesis includes the literature review of VRP, the problem descriptions of both VRPTWRB and SDVRPTWRB and solution approaches, implementation details of the algorithms to real-world applications and benchmark problems and conclusions. Chapter 2 examines both exact and heuristic algorithms for different VRP models from classical VRP to balanced VRP in the literature. Chapter 3 presents the mathematical model and structures of the heuristic method in detail of VRPTWRB.We also examine SDVRPTWRB in Chapter 3. In Chapter 4, we analyze the results obtained from the algorithms and present the details of our computational implementation. In the last chapter, we evaluate the outcomes and discuss future research opportunities.

Chapter 2

LITERATURE REVIEW

Dantzig and Ramser introduced the vehicle routing problem in 1959 as a truck dispatching problem. They developed the model as a generalization of traveling salesman problem (TSP) with an additional capacity constraint. To simplify the problem, they assumed only one type of product and one type of vehicle that has the same capacity [Dantzig and Ramser, 1959].

In general, VRP designs optimal sets of routes to provide service for a set of customers, using a set of vehicles; subject to side constraints. The objective of the problem can be to minimize: total cost of transportation, total number of vehicles, total time, total distance, or the weighted combination of them. Balinski and Quandt describe the total cost in terms of activities. They calculate the total cost using the weight of orders in the activities [Balinski and Quandt, 1964]. Side constraints can extend the problem in various directions from easiest capacity constraints to more complicated combination of constraints. Ghiani, Laporte, Musmanno state that common operational constraints are: the number of vehicles, demand- capacity constraint, duration of the route, time- windows, customer- specific vehicle constraint, and customer precedence relations [Ghiani et al., 2013]. We examine classical VRP with capacity constraints, VRP with time window constraints, and VRP with route balancing, in order.

Various exact and heuristic algorithms are developed to solve classical VRP models. We can classify exact algorithms in three main categories: direct tree research methods, dynamic programming, and integer linear programming. The lower bound assignment and a related branch-and-bound algorithm utilize the relationship between VRP and m-TSP. The latter problem comprises establishing a number of minimum cost routes starting and ending at depot while visiting each customer only once [Lenstra and Kan, 1975]. Another solution algorithm is developed based on k-degree center tree relaxation of fixed number of m-TSP for symmetric VRPs. The number of customers ranging between 10-25 can be successfully solved [Christofides et al., 1981]. Eilon, Watson-Gandy and Christofides propose a dynamic programming model to solve VRP with a fixed number of vehicles. They introduce a minimum cost recursion function to compute the optimal value [Burrows et al., 1972]. Christofides achieves an improved result of solving for 50 customers in a later study [Christofides, 1985]. Balinski and Quandt introduce a set partitioning and column generation algorithm for VRPs. The difficulties of this formulation are: the cost of computing routes and the risk of having a lot of binary variables. The problem can work for tightly constrained problems, since the feasible solutions are very few [Balinski and Quandt, 1964].

Heuristic algorithms for the VRP can be derived from the algorithms, that are developed for the traveling salesman problem (TSP) [Laporte, 1992]. Clarke and Wright algorithm solve VRPs with unlimited number of vehicles [Clarke and Wright, 1964]. The algorithm merges the customers to generate largest savings and continues until no improvement is possible. Variants of the method are proposed, such as inserting vehicle fixed costs and fleet sizes to the problem [Gaskell, 1967], [Yellow, 1970], [Paessens, 1988]. Wren and Holliday propose a sweep algorithm for VRPs with one or several depots, and vertices located in Euclidean plane represented with polar coordinates. The algorithm chooses a customer having the smallest angle and inserts the customer to a vehicle if it has available capacity. For each route generated, a TSP is solved to minimize the total distance traveled on routes [Wren and Holliday, 1972]. Gendreau, Hertz and Laporte develop a tabu search heuristic to construct a sequence of solutions and improvement steps [Gendreau et al., 1994]. If the routes generated by the algorithm are not feasible, a penalty in the objective function is measured to indicate the deviation from feasibility.

In VRPTW, each customer is associated with a travel cost, travel time, demand and time window $[e_i, l_i]$, where e_i and l_i represent earliest and latest acceptance times. The time windows might be assumed as hard or soft constraints, depending on the nature of the problem. We can classify the exact methods for the VRPTW in three categories; Lagrangian relaxation-based methods, column generation and dynamic programming [El-Sherbeny, 2010].

School bus routing and scheduling problem also considers time windows. Swersey and Ballard study the problem as routing and scheduling [Swersey and Ballard, 2008]. They assume that the routes are formed beforehand, hence only the scheduling problem is solved. A single bus is assigned to each route, pick up the students and arrive their school within a specified time window. The problem assumes that each bus is assigned to only one school. The objective is to minimize the number of school buses used. Swersey and Ballard solve the problem to optimality and formulate two integer programming models. They implement the formulations to the actual data of a region in the USA. The method yields satisfying results; they reduce the number of buses needed by 25 %.

The heuristic algorithms are widely used to solve VRPTW. Baker and Schaffer design route building heuristics, which is an extension of Clarke and Wright saving heuristic of VRP [Baker and Schaffer, 1986]. They define the savings as a combination of distance and time. Solomon defines a similar heuristic; however, time is not included in the savings function [Solomon, 2008]. Landeghem develops a saving heuristic to measure the time between customers [Landeghem, 1988]. These heuristics have time complexity. r-Opt heuristic replaces r customers by other r customers. 2-Opt or 3-Opt can solve the problem, despite the risk of time window violation. Potvin and Rousseau develop 2-Opt and r-Opt algorithms to solve such problems [Potvin and Rousseau, 1995]. Metaheuristic methods are also used in solving VRPTW. These methods include simulated annealing [Chiang and Russell, 1996], tabu search [Taillard et al., 1997] and genetic algorithm [Thangiah, 1995]. Genetic algorithm is one of the favorable approximate approaches that is frequently used in solving VRPTW.

So far, we discuss the basic VRP concept and corresponding solution methods. The aim is to extend the classical VRP to a more realistic problem; route balancing in VRPTW. VRPRB is considered as load balancing or distance balancing in the literature. Lee and Ueng study VRP with load balancing [Lee and Ueng, 1999]. Their objective is to minimize total time and balance the routes, simultaneously. They refer to load balance as workload equity among employees. The problem consists of a single service station (depot) with multiple vehicles. They solve the problem by a heuristic algorithm with two objectives: the shortest travel path and load balance amongst drivers. The study's main contribution is to improve the workload equity amongst drivers. However, the study does not include the time window constraints of the customers; the total available time of the vehicles are the only time limits considered in their model.

Lacomme et al. compare statistically different objective functions for the VRPRB [Lozano et al., 2018]. All the routes generated should be as similar as possible in the problem. They compare seven different objective functions. Max- min aims to minimize the difference between minimum and maximum tour length. Min- max tries to minimize the maximum tour length. Third objective function, Var, minimizes the variance of tour lengths. Rel function minimizes difference between the tour length for the selected route, and the minimum tour length over all routes. Mean Absolute Deviation (MAD) function calculates the difference between tour length of each route and, the mean of tour lengths of all routes generated. Gini function minimizes the inequality between tour lengths. Experiments reveal that Var, MAD and Gini functions are superior. This study only considers balanced total load on vehicles or total length of routes. Time windows of the customers are not included in any of the models. Matl, Harti and Vidal consider both route balancing and time windows. They have a detailed review paper on workload equity in VRP. They discuss various equity metrics (min- max, lexicographic min- max, range, mean absolute deviation, standard deviation, and gini coefficient), solution methods, and objectives [Matl, P., Hartl, R. F., & Vidal, 2017]. The results reveal that monotonic objective functions are convenient for VRPRB. This study has a sophisticated perspective in examining the problem and addresses useful ideas.

Chen and Chen also study VRP with load balancing and time windows in distribution [Chen and Chen, 2008]. The objectives are: to minimize the number of vehicles, obtain the minimum distribution distance, balance the load for vehicles while obeying the time windows. They assign a pre- determined average load level for each vehicle to prevent the imbalance. They construct a two-stage solution approach. In the first stage, they apply a particle swarm optimization method. The second stage assures the balance of the routes. The method is suitable to solve medium sized problems. Zhou et al. propose a bi-objective VRPRB based on a genetic algorithm [Zhou et al., 2013]. The objectives of the study are to minimize the total distance traveled and the distance imbalance of the vehicles. They validate the algorithm on Solomon test problems; and conclude that the distance balance was improved in all test problems. They also state that the route balance is related with balance in the number of vehicles visited by each vehicle, load of each vehicle, required completion time of the route, required waiting time of the route, delayed time of the route, and distance traveled by each vehicle.

Belfiore and Yoshizaki propose a heuristic method for the mixed- fleet VRPTW with split deliveries [Belfiore and Yoshizaki, 2013]. This study does not consider the route balance in VRP; however, the foundation of our proposed algorithm is based on it. The model involves different fleet types, different fleet sizes and allows split delivery. They adapt scatter- search method to solve the problem. The model minimizes the total number of vehicles and total distance. Scatter search algorithm has three main steps: diversification generation method, improvement method, and reference set building. A pre- defined number of initial solutions are generated, in which all the customers are assigned to available vehicles. Five improvement steps are applied to initial solutions. Among the improved solutions, the algorithm chooses several of them to be in the set of the best solutions. Among the unchosen improved solutions, several of them are chosen to be in the set of the worst solutions. The best and worst solutions form the reference set. Finally, the algorithm combines the solutions in the reference set, and the resulting set of routes is formed. The algorithm is applied to Solomons problem sets. The experiments reveal that for certain problem sets, the algorithm achieves satisfying results. The computation time is significantly lower than the other proposed methods in literature.

Chapter 3

VEHICLE ROUTING PROBLEM WITH TIME WINDOWS AND ROUTE BALANCE

There are various VRPTW models developed to generate routes in which all nodes are visited within a specified time windows, while satisfying vehicle capacity and time window restrictions. The routes are generated to minimize the total distance traveled, total time spent, total cost incurred or a combination of these. However, there may not be an equivalence or similarity between the routes generated; a vehicle may leave the depot early in the morning whereas another one may leave in the afternoon. Moreover, a vehicle may return to the depot after visiting all of the nodes assigned to it in afternoon while another vehicle may return at midnight. This imbalance between the routes also affects the working conditions of drivers. This is the main motivation for us to develop a mathematical model and algorithms that solve the VRPTW with balanced routes.

There are some models developed for VRPTWRB. But they usually consider balance of load, total time or total distance. The difference of our model is that, we seek a balance of start and end times of the routes; we are trying to make the time that vehicles leave the depot and the time the vehicles return to the depot as close as possible. Therefore, there will not be large gaps in working time of the drivers serving different routes.

The vehicles start their routes from a depot and return to the depot at the end of the day. The customers have earliest and latest acceptance times and they should be served within those time windows. The fleet of vehicles can vary in capacities and sizes.

Section 3.1 presents the mathematical formulation of the VRPTWRB model. Section 3.2 is dedicated to the heuristic algorithm that is developed to solve the large size problems. In Section 3.3, we define a split delivery extension of VRPTWRB and discuss the mathematical and heuristic models of this problem.

3.1 Problem Description

VRPTWRB generates minimum number of routes while minimizing the total travel time and balance the routes in terms of route completion time. This problem is an extension of the vehicle routing problem with time- windows. This MIP model is based on the work of De Santiago et al [De Santiago et al., 2013]. We consider this model with two different objective functions: i) minimizing the total travel time, ii) minimizing the difference between the total travel times of the longest and the shortest route.

We have some assumptions in the problem. First, the total demand of customers should be less than or equal to the total capacity. Second, the time windows of customers should be within the working hours of the depot. The service time of a customer cannot be greater than the duration of the time window of that customer. We also assume that there is only one type of good to be delivered and all types of vehicles can visit all customers. Customers to be served are determined prior to route generation. Therefore, the number of customers and their demands are deterministic. The fleet size is fixed and the capacities of the vehicles are known. The fixed and variable cost of vehicles are not considered in the model. The service time of customers are known. The travel times between customers are asymmetric, reflecting a typical urban setting. The speed of the vehicles are assumed to be fixed and same. Thus, travel times does not differ among the vehicles.

For the depot, it is assumed that the service time is zero and all vehicles depart the depot at time zero. The latest return time to depot is considered as the maximum tour length of a vehicle, therefore, the length of a route should not exceed that value.

The model is stated on the next page:

Sets:

V	set of nodes = 0,1,2,...,N
I	set of customers = 1,2,...,N
K	set of vehicles= 1,2,...,K

Parameters:

d_i	$i \in I$	demand of customer i
t_{ij}	$i, j \in V$	travel time between node i and node j
s_i	$i \in I$	service time of customer i
a_i	$i \in I$	earliest acceptance time of customer i
b_i	$i \in I$	latest acceptance time of customer i
q_k	$k \in K$	maximum capacity of vehicle k
M		A large number

Variables:

x_{ijk}	1 if arc (i, j) is used by vehicle k, 0 otherwise	$i, j \in V, k \in K$
y_{ik}	1 if vehicle k arrives before the earliest acceptance time of customer i, 0 otherwise	$i \in I, k \in K$
r_{ijk}	waiting time of vehicle k on arc (i, j) before serving customer j,	$i, j \in V, k \in K$
e_{ik}	the time that vehicle k starts serving customer i,	$i \in I, k \in K$
v_{ik}	the time vehicle k arrives at customer i,	$i \in I, k \in K$
L_{max}	total travel time of longest route	
L_{min}	total travel time of shortest route	

Objectives:

$$z_1 = \sum_{k \in K} \sum_{i \in I} \sum_{j \in I: i \neq j} t_{ij} x_{ijk},$$

$$z_2 = L_{max} - L_{min}$$

$$\bar{z} = z_1 \ \ or \ \ z_2$$

$$\text{minimize} \quad \bar{z} \tag{3.1}$$

$$\text{subject to} \quad \sum_{k\in K}\sum_{i\in V} x_{ijk} = 1 \qquad \forall j \in V\backslash\{0\} \tag{3.2}$$

$$\sum_{j\in V} x_{jik} - \sum_{h\in V} x_{ihk} = 0 \qquad \forall i \in V, k \in K \tag{3.3}$$

$$\sum_{i\in I}\sum_{j\in I} d_i x_{ijk} \leq q_k \qquad \forall k \in K \tag{3.4}$$

$$a_i \leq e_{ik} \leq b_i \qquad \forall i \in I, k \in K \tag{3.5}$$

$$e_{ik} + s_i + t_{ij} - M(1 - x_{ijk}) \leq e_{jk} \qquad \forall k \in K, i, j \in I \tag{3.6}$$

$$\begin{aligned} b_i - a_j \geq\ & x_{ijk}(b_i + s_i + t_{ij} - a_j) + x_{jik}(b_i + s_j - t_{ji} - a_j) \\ & + e_{ik} - e_{jk} + r_{ijk} - r_{jik} \qquad \forall k \in K, i, j \in I \end{aligned} \tag{3.7}$$

$$e_{ik} = v_{ik} + \sum_{j\in V: i\neq j} r_{jik} \qquad \forall i \in I, k \in K \tag{3.8}$$

$$0 \leq r_{ijk} \leq b_0 x_{ijk} \qquad \forall k \in K, i, j \in I \tag{3.9}$$

$$a_i \leq v_{ik} + b_0 y_{ik} \leq a_i + b_0 \qquad \forall i \in I, k \in K \tag{3.10}$$

$$\sum_{j\in V} r_{ijk} - b_0 y_{ik} \leq 0 \qquad \forall i \in I, k \in K \tag{3.11}$$

$$e_{ik} - (b_i - a_i) y_{ik} \leq b_i \qquad \forall i \in I, k \in K \tag{3.12}$$

$$v_{ik} \geq t_{0i} x_{0ik} \qquad \forall i \in I, k \in K \tag{3.13}$$

$$e_{ik} \leq (t_{0i} - b_i) x_{0ik} + r_{0ik} + b_i \qquad \forall i \in I, k \in K \tag{3.14}$$

$$e_{ik} - v_{jk} + (b_0 + s_i + t_{ij}) x_{ijk} \leq b_0 \qquad \forall k \in K, i, j \in I \tag{3.15}$$

$$L_{max} \geq e_{ik} + s_i + t_{i0} x_{i0k} \qquad \forall i \in I, k \in K \tag{3.16}$$

$$L_{min} \leq e_{ik} + s_i + t_{i0} x_{i0k} + (1 - x_{i0k}) b_0 \qquad \forall i \in I, k \in K \tag{3.17}$$

$$y_{ik} \leq x_{ijk} \qquad \forall i, j \in I, k \in K \tag{3.18}$$

$$x_{ijk} \in \{0, 1\} \qquad \forall k \in K, i, j \in I \tag{3.19}$$

$$y_{ik} \in \{0, 1\} \qquad \forall i \in I, k \in K \tag{3.20}$$

$$r_{ijk} \geq 0 \qquad \forall k \in K, i, j \in I \tag{3.21}$$

$$e_{ik} \geq 0, v_{ik} \geq 0 \qquad \forall i \in I, k \in K \tag{3.22}$$

$$L_{max} \geq 0, L_{min} \geq 0 \tag{3.23}$$

The objective functions z_1 and z_2 define total travel time of the routes and the difference between the longest and shortest routes in terms of travel time, respectively. The objective of the model (3.1) is to minimize z_1 or z_2, so that a balance between the routes are satisfied. Constraint (3.2) ensures that each customer is served by exactly one vehicle. Constraint (3.3) is the flow balance constraint, which ensures that each incoming arc must go out. The capacity constraint of each vehicle is defined by (3.4). Constraints (3.5) and (3.6) are the time windows constraints and ensure that the time that service starts at customer i on route k should be between the earliest and latest time windows of the customer. If customer i is not carried by vehicle k, then e_{ik} gets a dummy value of a_i and constraints (3.5) and (3.6) will be redundant. Constraint (3.7) is required to compute the time at which service starts at customer j after customer i. If $x_{ijk} = 1$ and $x_{jik} = 0$, the constraint ensures that customer j will be served after customer i. Likewise, if $x_{jik} = 1$ and $x_{ijk} = 0$, the constraint ensures that customer i will be served after customer j. If both x_{jik} and x_{ijk} are zero, then the constraint becomes $b_i - a_j \geq e_{ik} - e_{jk}$. Since the constraint (3.5) enforces e_{ik} to be at most b_i and similarly e_{jk} to be at least a_j, the maximum difference that $e_{ik} - e_{jk}$ can get is $b_i - a_j$. Therefore, when both x_{jik} and x_{ijk} are zero, constraint (3.7) will be redundant. Constraint (3.8) ensures that the time that service starts is equal to the sum of the times that the vehicle arrives at the customer and the waiting time before the service starts. If customer i is not in the vehicle k, the constraint becomes $e_{ik} = v_{ik}$. Therefore, v_{ik} also gets the dummy value of a_i. Constraint (3.9) computes r_{ijk} only if the arc (i, j) is used. Otherwise, r_{ijk} will be zero and the constraint will be redundant. Constraint (3.10) ensures that if a vehicle arrives before the earliest time, a_i, then y_{ik} must be 1. If that is the case, then the right hand side of the inequality is redundant. If the vehicle arrives after the earliest time of the customer, then y_{ik} must be 0, the right- hand side of the inequality will naturally hold, and the left- hand side of the inequality will be redundant. If customer i is not in the vehicle k, the constraint becomes $a_i \leq a_i \leq a_i + b_0$ and will be redundant. Constraint (3.11) enforces the vehicles to start service as soon as possible. If customer i is not in the vehicle k, $\sum_{j \in V} r_{ijk}$ and y_{ik} are zero for all $i, j \in V$ and $k \in K$. Therefore, the constraint becomes redundant. Constraint (3.12) also ensures the service to start as soon as possible, while enforcing the service start time to be between the time- window of the customer. If $x_{ijk} = 0$, the constraint becomes $a_i \leq b_i$ and will be redundant. Constraint (3.13) ensures the arrival time of the vehicle to the first customer to be greater than the travel time between depot and the first customer, if arc (i, j) is used. Otherwise, v_{ik} will be trivially equal to a_{ik}. In constraint (3.14), if customer i is the first customer in route k, the service time must be less than or equal to the sum of travel time from depot and the waiting time. If customer i is not the first customer, then the constraint will be redundant and only ensures the service start time to be before the latest time of the customer. If customer i is not in the vehicle k, the constraint becomes $a_i \leq b_i$ and will be redundant. Constraint (3.15) is defined for precedence relationships between customers i and j. If arc (i, j) is used, then the arrival time of the vehicle to customer j should be after serving customer i plus travel time between customers i and j. Otherwise, the constraint becomes redundant and no precedence relationship is considered. Constraints (3.16) and (3.17) are defined for computing the longest and shortest routes in terms of travel time. For calculating the longest route, if customer i is the last one in route k, then the constraint (3.16) ensures that the length of the tour is at least the sum of the instant at which service starts, the service time of customer i and the travel time to depot from the customer. If i is not the last customer in the route, then the constraint becomes redundant. Likewise, if customer i is the last one in route k, then L_{min} is at most the sum of the instant at which service starts, the service time of customer i and the travel time to depot from the customer. Else, then L_{min} is at most a very large value, which makes the constraint redundant. For constraints (3.16) and (3.17), if customer i is not the first customer, then the constraints become $L_{max} \geq a_i + s_i$ and $L_{min} \leq a_i + s_i + b_0$, respectively and will be redundant. Therefore, constraints (3.16) and (3.17) are meaningful for the last customers in routes, and choose the maximum and minimum travel times among the routes, respectively. The last functional constraint, (3.18), ensures that variable

y_{ik} takes value 1 only if the the arc (i, j) is used. Constraints (3.19)- (3.23) indicate the variables x_{ijk} and y_{ik} are binary, and r_{ijk}, e_{ik},u_{ik},v_{ik},L_{max} and L_{min} can take non-negative values.

3.2 Heuristic Algorithm

In the previous section, we give the mathematical formulation of the VRPTWRB. Since the model is NP-hard, it cannot solve large problems in polynomial time. In this section, we discuss a heuristic algorithm that solves the problem with large size instances.

Our algorithm is based on the scatter search (SS) algorithm which was first introduced by Glover[Glover, 1977]. Marti, Laguna and Glover state that SS algorithm can be applied to integer programming problems. SS algorithm generates solutions by using a weighted linear combination of a set of reference solutions [Martí et al., 2006]. The five well- known methods of the algorithm are: diversification generation, improvement, reference set update, subset generation and solution combination methods [Glover, 1998]. Figure [3.1] illustrates the design of the SS algorithm.

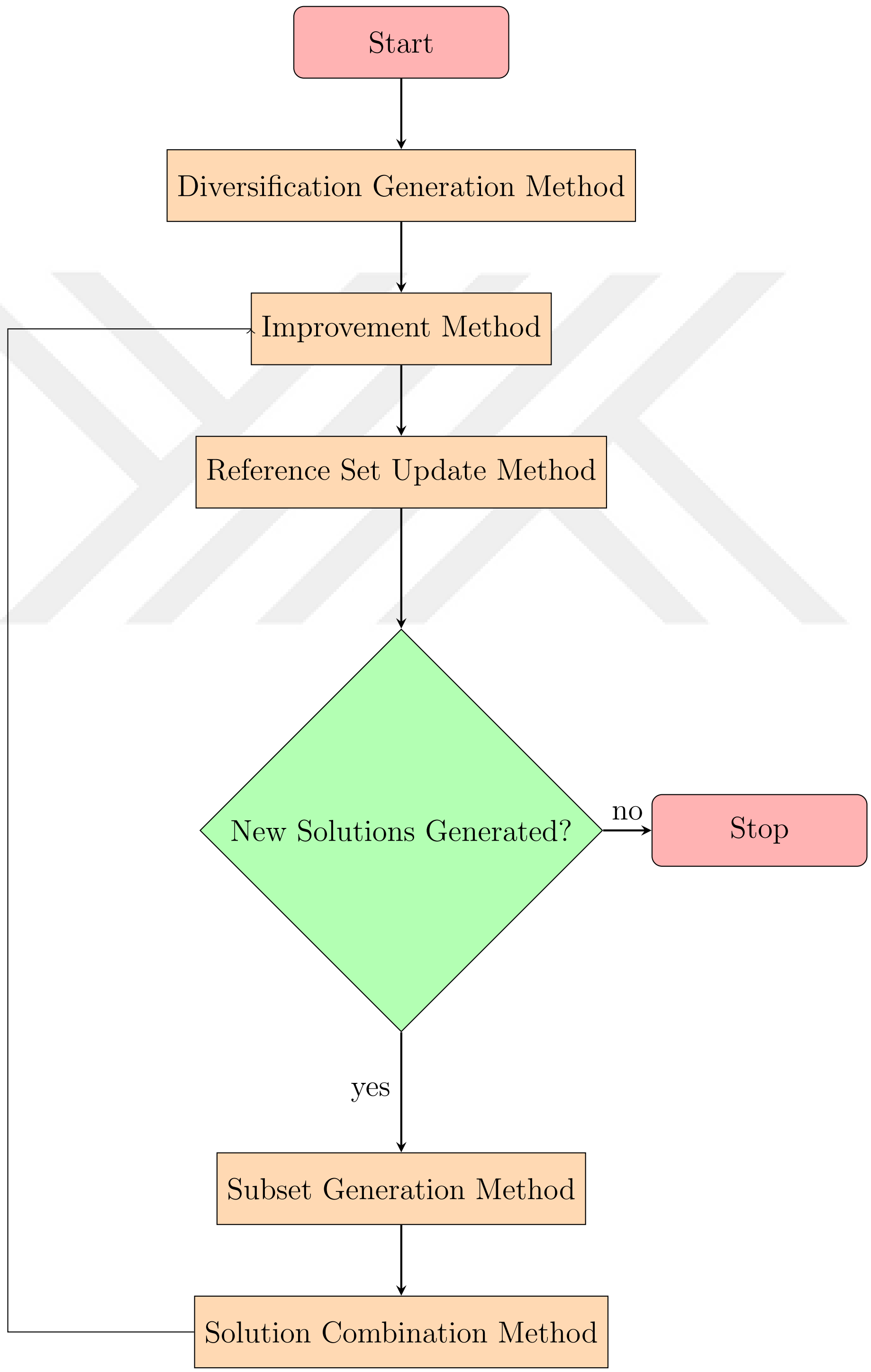

Figure 3.1: The Scatter Search Algorithm Design (adapted from [Glover, 1998])

The parameters of the SS algorithm are:

$PSize$	The size of the set of diverse solutions generated in diversification generation method
b	The size of the reference set (RefSet)
b_1	The size of the subset of best solutions in RefSet
b_2	The size of the subset of diverse solutions in RefSet
$IterNo$	Maximum number of iterations

Algorithm 1 demonstrates SS method based on the work of Alegre, Laguna and Pacheco[Alegre et al., 2007]. We use a similar design of SS method to solve the VRPTWRB. We consider multiple vehicle types that have different capacities and fixed costs, and customers with time windows. The objective is to find a solution that yields the minimum travel time and minimum number of vehicles used while having a balance among the routes. The inputs of the algorithm are demand, service time, earliest and latest acceptance times of customers, distance and travel time matrices between the depot and the customers, and capacities of the vehicles.

Our algorithm is superior than the mathematical model that is developed in Section 3.1. In the mathematical model, we are given pre-defined distance and travel time matrices and find the optimal set of routes using these matrices. However, in the heuristic algorithm, we are able to calculate the distance and travel time between two nodes dynamically. During the diversification generation and improvement methods, we import the dynamic travel time and distance values between the two nodes from Google Distance Matrix API. With this method, we can build real-time routes and obtain more accurate solutions rather than the solutions obtained from the mathematical model.

Algorithm 1: Scatter Search Algorithm

```
Number Of Iterations=0
while Number Of Iterations < IterNo do
    Step 1:  Generate Psize number of solutions with diversification
             generation method
    Step 2:  Improve the solutions generated in Step 1 using Improvement
             Methods
    Step 3:  Build the reference set of size b from the set of improved
             solutions in Step 2. Build subsets of best and diverse solutions
    while New solutions are added to RefSet do
        Step 4:  Apply solution combination method
        Step 5:  Apply improvement method for combined solutions
        Step 6:  Update RefSet, best solutions set and diverse solutions set
                 considering the solutions in step 5
    end
    Number Of Iterations = Number Of Iterations + 1
end
```

3.2.1 Diversification Generation Method

In the first step, diversification generation method generates a set of solutions for $PSize$ number of elements. This step is the initialization of the algorithm. The customers are assigned to vehicles without violating the time-window and capacity restrictions. The vehicles are randomly selected. However, the customer selection is made either using nearest neighbor (NN) algorithm, or using a priority, which is assigned to each customer according to its time-windows. Nearest neighbor is one of the well-known construction heuristics to solve VRP. It constructs a route by assigning an unselected customer to a vehicle and in each iteration it selects the customer, among the set of unselected customers, which is nearest to the last customer inserted in the route [Bräysy and Gendreau, 2005].

Another way to construct initial solutions is to assign a priority coefficient to each customer. The calculation of priorities is stated below:

$$\frac{(ET_i - ET_{min}) + (LT_{max} - LT_i)}{LT_i - ET_i}.$$

ET_i and LT_i are defined as earliest and latest acceptance times of each customer, respectively. ET_{min} and LT_{max} are the minimum and maximum acceptance times among all customers. We choose the unselected customer with the highest priority coefficient and assign it to the selected vehicle. We repeat this procedure until the capacity or time- window constraint is violated or all the customers are assigned to a vehicle.

In this step, the time balance of the routes generated are also considered. We define the parameters: total service time (TST), single route length (SRL), minimum number of vehicles used (N), and the minimum travel time from depot (MD). TST is the sum of service time of all customers. SRL is calculated using NN algorithm [Bräysy and Gendreau, 2005], so that it will give the total travel time of a single route that contains all the customers. N is found by dividing total demand of customers to the capacity of the largest vehicle. Finally, MD is the sum of the travel times from the depot of the closest N customers. We calculate an average total time (ATT) of a route using the following formula:

$$ATT = \frac{TST + SRL + MD}{N}.$$

We also define a balance fraction (BF) parameter, which is between 0 and 1. A route is constructed if the total route length is between $(ATT - ATT * BF)$ and $(ATT + ATT * BF)$.

Choosing between NN and priority assignment depends on the size and type of the data. For some data sets NN gives better results, whereas priority assignment performs better for others. Therefore, the algorithm initializes the diverse solutions by using both NN algorithm and priority assignment. The set of solutions prepared by both methods are compared and the superior set is chosen. Then, we proceed to the next step of the algorithm.

3.2.2 Improvement Methods

After the initial solutions are generated, a sequence of improvement methods are applied. These methods are based on the work of Belfiore and Yoshizaki [Belfiore and Yoshizaki, 2013]. The improvement steps are: swap of customers in the same route, elimination and combination of the routes and insertion. These improvement methods are useful for decreasing the total time spent in a route and decreasing the number of vehicles used.

Swap

In the swap phase, the aim is to reduce the total time of each route. This phase is based on 2-opt algorithm [Croes, 1958]. For each route, two customers in that route are selected randomly. The positions of these customers are exchanged and new total time of the route is calculated. If there is a decrease in the total time and the time window constraints are satisfied, then the route is updated. If there is no improvement, there is no update in the route. The procedure continues for each route until no other exchange is possible.

Route Elimination and Combination

In route elimination and combination step, the aim is to reduce the number of vehicles used. In some cases, in the initialization step, the occupancy of vehicles may be very low. Sometimes, only one or two customers may be assigned to the vehicle. In such cases, it may be possible to eliminate that route and allocate another vehicle to the customer(s) in the eliminated route. In this step, we calculate the occupancy levels of each vehicle. If a vehicle has empty capacity, we select another route and try to merge these two routes. If the capacity and time-window constraints are satisfied and total length of the route is reduced, we choose the new route. In this step, we try all possible pair of routes to make an improvement. For example, we select route R_i and calculate the idle capacity of vehicle i. Among the routes $R_1, R_2, ..., R_{i-1}, R_{i+1}, ..., R_m$, we choose a route R_j. If total demand in route R_j does not exceed the available capacity in route R_i, we merge it with the route R_i. If the merge is feasible, we update the route R_i and eliminate the route R_j and continue with the next route. If the merge is not feasible, we choose another route from $R_1, R_2, ..., R_{i-1}, R_{i+1}, ..., R_m$. If there is no available routes that can be merged with route R_i, we continue to select a new route from the list, $R_1, R_2, ..., R_i, ..., R_m$.

Insertion

In the last improvement step, insertion, we try to remove a customer from its current route and insert it to another route. The aim is to reduce the total travel time and the total number of vehicles used. In this step, we choose the routes one by one from the list of routes, $R_1, R_2, ..., R_i, ..., R_m$. We choose the customer, c_i that is farthest from the depot in route R_i. Then, we select a candidate route, R_j, from the list of routes, $R_1, R_2, ..., R_{i-1}, R_{i+1}, ..., R_m$ that has available capacity for c_i. We insert c_i in all possible positions that satisfies the constraints in route R_j and calculate the total travel time of each scenario. We apply the same procedure to each $R_1, R_2, ..., R_{i-1}, R_{i+1}, ..., R_m$ for inserting c_i. Then, the route that satisfies all the restrictions and gives the minimum total travel time is selected among the candidate route- position pairs. Routes R_i and R_j are updated accordingly. The procedure continues with choosing a route from $R_1, R_2, ..., R_i, ..., R_m$ and a new customer that is farthest to the depot in that route.

3.2.3 Reference Set Update Method

Among the $PSize$ number of improved solutions, b_1 and b_2 of them are selected to be in the best and diverse solutions sets, respectively. The selection criteria is based on total travel time. Therefore, the improved solutions are sorted in ascending order according to the total travel time. Then, first b_1 of the sorted solutions are selected and inserted to the best solutions set, and last b_2 of them are selected and inserted to the diverse solutions set.

3.2.4 Combination Method

The next step of the algorithm is to combine the best and diverse solutions. To do this, we combine the solutions within the best set. We also combine the solutions in the best and diverse sets. Thus, the total number of combined solutions can be calculated as follows:

$$\frac{b_1(b_1-1)}{2}+b_1b_2$$

For each combination, we calculate the number of common customers that are served in the same route. For example, let S^1 and S^2 be two solutions that contain s_1 and s_2 routes, respectively. We combine each route S_i^1 with each route S_j^2 and calculate the number of customers that are both in S_i^1 and S_j^2, where $i = 1, ..., s_1$ and $j = 1, ..., s_2$. Apart from Belfiore and Yoshizaki, in our combination procedure, we determine the maximum number of common customers in route combinations and generate a route that contains them. We choose the vehicle randomly between the vehicles used in routes S_i^1 and S_j^2. We then insert remaining customers to that route using NN method, if there is enough empty capacity. If there is no capacity left, the remaining customers are assigned to a route using the initialization method.

After the combined solution is generated, the improvement methods are applied to ensure that the total travel time and total number of vehicles used are reduced. As the last step, the algorithm solves TSP for the routes generated.

The algorithm continues until maximum number of allowed iterations, $IterNo$, is reached. In the final step, there are $IterNo$ number of solutions generated in each iteration. The best solution, that has the minimum total travel time and minimum number of vehicles, is selected among the candidate solutions. We test our algorithm with different $PSize$, b_1, b_2 and $IterNo$ values. For different data sets, the parameters can take different values. A detailed explanation is given in Chapter 4.

3.3 Split Delivery in Vehicle Routing Problem with Time Windows and Route Balance

Split delivery vehicle routing problem (SDVRP) aims to find the optimal set of routes that serve all customers while ensuring the capacity limitations, with the flexibility of splitting customer demand. Therefore, the customers whose demand are greater than the capacity of the vehicles can be visited more than once [Dror and Trudeau, 1990]. In this section, we extend SDVRP by adding the time window and route balance constraints (SDVRPTWRB).

This section involves the mathematical formulation of SDVRPTWRB. This problem is also NP-hard and cannot solve large size problems[Archetti et al., 2006]. Therefore, a heuristic algorithm is designed to solve large problems in Section 3.3.1.

3.3.1 Problem Description

In this section, we introduce the mathematical model that solves SDVRPTWRB. The problem is an extension of VRPTWRB. The objective is to find an optimal set of routes that minimizes the total travel time while ensuring capacity, time window and route balance constraints. We allow split delivery to customers in the model, so that more than one vehicle can visit a customer. We have the same assumptions with VRPTWRB for the problem to be feasible.

The complete model is stated below.

Sets:

V set of nodes = 0,1,2,...,N

I set of customers = 1,2,...,N

K set of vehicles= = 1,2,...,K

Parameters:

d_i	$i \in I$	demand of customer i
t_{ij}	$i, j \in V$	travel time between node i and node j
s_i	$i \in I$	service time of customer i
a_i	$i \in I$	earliest acceptance time of customer i
b_i	$i \in I$	latest acceptance time of customer i
q_k	$k \in K$	maximum capacity of vehicle k

Variables:

x_{ijk}	1 if arc (i, j) is used by vehicle k, 0 else	$i, j \in V, k \in K$
y_{ik}	1 if vehicle k arrives before the earliest acceptance time of customer i, 0 else	$i \in I, k \in K$
r_{ijk}	waiting time of vehicle k on arc (i, j) before serving customer j,	$i, j \in V, k \in K$
e_{ik}	the time that vehicle k starts serving customer i,	$i \in I, k \in K$
w_{ik}	fraction of demand of customer i delivered by vehicle k,	$i \in I, k \in K$
v_{ik}	the time vehicle k arrives at customer i,	$i \in I, k \in K$
L_{max}	total travel time of longest route	
L_{min}	total travel time of shortest route	

Objectives:

$z_1 = \sum_{k \in K} \sum_{i \in I} \sum_{j \in I: i \neq j} t_{ij} x_{ijk},$

$z_2 = L_{max} - L_{min}$

$\bar{z} = z_1 \;\; or \;\; z_2$

$$\text{minimize} \quad \bar{z} \tag{3.24}$$

$$\text{subject to} \quad \sum_{k \in K} \sum_{i \in V} x_{0jk} = 1 \qquad \forall j \in I \tag{3.25}$$

$$\sum_{k \in K} w_{ik} = 1 \qquad \forall i \in I \tag{3.26}$$

$$w_{ik} \leq \sum_{j \in V : i \neq j} x_{jik} \qquad \forall i \in I, k \in K \tag{3.27}$$

$$\sum_{i \in I} d_i w_{ik} \leq q_k \qquad \forall k \in K \tag{3.28}$$

$$\sum_{j \in V : i \neq j} x_{jik} - \sum_{h \in V : i \neq h} x_{ihk} = 0 \qquad \forall i \in V, k \in K \tag{3.29}$$

$$a_i \leq e_{ik} \leq b_i \qquad \forall i \in I, k \in K \tag{3.30}$$

$$e_{ik} + s_i + t_{ij} - M(1 - x_{ijk}) \leq e_{jk} \qquad \forall k \in K, i, j \in I : i \neq j \tag{3.31}$$

$$\begin{aligned} b_i - a_j \geq\ & x_{ijk}(b_i + s_i + t_{ij} - a_j) + x_{jik}(b_i + s_j - t_{ji} - a_j) \\ & + e_{ik} - e_{jk} + r_{ijk} - r_{jik} \qquad \forall k \in K, i, j \in I \end{aligned} \tag{3.32}$$

$$e_{ik} = v_{ik} + \sum_{j \in V : i \neq j} r_{jik} \qquad \forall i \in I, k \in K \tag{3.33}$$

$$0 \leq r_{ijk} \leq b_0 x_{ijk} \qquad \forall k \in K, i, j \in I : i \neq j \tag{3.34}$$

$$a_i \leq v_{ik} + b_0 y_{ik} \leq a_i + b_0 \qquad \forall i \in I, k \in K \tag{3.35}$$

$$\sum_{j \in V : j \neq i} r_{ijk} - b_0 y_{ik} \leq 0 \qquad \forall i \in I, k \in K \tag{3.36}$$

$$e_{ik} - (b_i - a_i) y_{ik} \leq b_i \qquad \forall i \in I, k \in K \tag{3.37}$$

$$v_{ik} \geq t_{0i} x_{0ik} \qquad \forall i \in I, k \in K \tag{3.38}$$

$$e_{ik} \leq (t_{0i} - b_i) x_{0ik} + r_{0ik} + b_i \qquad \forall i \in I, k \in K \tag{3.39}$$

$$e_{ik} - v_{jk} + (b_0 + s_i + t_{ij}) x_{ijk} \leq b_0 \qquad \forall k \in K, i, j \in I : i \neq j \tag{3.40}$$

$$L_{max} \geq e_{ik} + s_i + t_{i0} x_{i0k} \qquad \forall i \in I, k \in K \tag{3.41}$$

$$L_{min} \leq e_{ik} + s_i + t_{i0} x_{i0k} + (1 - x_{i0k}) b_0 \qquad \forall i \in I, k \in K \tag{3.42}$$

$$y_{ik} \leq x_{ijk} \qquad \forall i, j \in I, k \in K \tag{3.43}$$

$$x_{ijk} \in 0, 1 \qquad \forall k \in K, i, j \in I : i \neq j \tag{3.44}$$

$$y_{ik} \in 0, 1 \qquad \forall i \in I, k \in K \tag{3.45}$$

$$r_{ijk} \geq 0 \qquad \forall k \in K, i, j \in I : i \neq j \tag{3.46}$$

$$e_{ik} \geq 0, v_{ik} \geq 0 \qquad \forall i \in I, k \in K \tag{3.47}$$

$$L_{max} \geq 0, L_{min} \geq 0 \tag{3.48}$$

The objective function is the same as the model in VRPTWRB. Constraint (3.25) ensures that each vehicle leaves the depot. Constraint (3.26) states that each customer's total demand will be fulfilled. Constraint (3.27) guarantees that if a customer i is served by vehicle k, then the demand of the customer will be fulfilled. Constraint (3.28) ensures that the vehicle capacity constraints will not be violated. Constraints (3.29) - (3.48) are the same constraints in the mathematical model of VRPTWRB.

3.3.2 Heuristic Algorithm

The mathematical model developed in section 3.1.1 can solve only small size problems. Therefore, we developed a heuristic algorithm which is also based on the work of Belfiore and Yoshizaki [Belfiore and Yoshizaki, 2013] and an extension of the algorithm in section 3.2. Since we allow split delivery, we can reduce the total number of vehicles used.

We consider multiple vehicle types that have different capacities and fixed costs. The objective is to find a solution that yields the minimum travel time and minimum number of vehicles used while creating a balance among the routes. The inputs of the algorithm are demand, service time, earliest and latest acceptance times of customers, distance and travel time matrices between the depot and the customers, and vehicle capacities.

The parameters of the algorithm is same as the algorithm in Section 3.2 and the design of the algorithm is the same with Figure 3.1. In the split delivery scatter search based heuristic algorithm, there are two additional improvement steps, reallocation and route addition methods. The steps of the developed algorithm can be seen in Algorithm 2.

Algorithm 2: Scatter Search Algorithm for Split Delivery

Number Of Iterations=0

while *Number Of Iterations* $<$ *IterNo* **do**

  Step 1: Generate Psize number of solutions with diversification generation method

  Step 2: Apply swap method to improve the solutions generated

  Step 3: Apply reallocation method to improve the solutions generated

  Step 4: Apply elimination and combination method to improve the solutions generated

  Step 5: Apply insertion method to improve the solutions generated

  Step 6: Apply route addition method to improve the solutions generated

  Step 7: Build the reference set of size b from the set of improved solutions in Step 2. Build subsets of best and diverse solutions

  while *New solutions are added to RefSet* **do**

    Step 8: Apply solution combination method

    Step 9: Apply improvement method for combined solutions

    Step 10: Update RefSet, best solutions set and diverse solutions set considering the solutions in step 9

  end

  Number Of Iterations = Number Of Iterations + 1

end

Diversification Generation Method

In the first step, diversification generation method generates a set of initial solutions for $PSize$ number of elements. The customers are assigned to vehicles without violating the time- window and capacity restrictions. The vehicles are randomly selected. However, the customer selection is made either using nearest neighbor (NN) algorithm, or using a priority, which is assigned to each customer according to its time-windows. Nearest neighbor is one of the well- known construction heuristics to solve VRP. It constructs a route by assigning an unrouted customer to a vehicle and in each iteration it selects the customer, among the set of unrouted customers, nearest to the last customer inserted in the route [Bräysy and Gendreau, 2005]

Another way to construct initial solutions is to assign a priority coefficient to each customer. The calculation of the priorities is stated below:

$$\frac{(ET_i - ET_{min}) + (LT_{max} - LT_i)}{LT_i - ETi}.$$

ET_i and LT_i are defined as earliest and latest acceptance times of each customer, respectively. ET_{min} and LT_{max} are the minimum and maximum acceptance times among all customers. We choose the unrouted customer with the highest priority coefficient and assign it to the selected vehicle.

We consider split delivery in this step. If the demand of the selected customer is greater than the remaining capacity of a vehicle, we split the demand and fill the vehicle with the maximum allowable demand. Thus, a customer may split into more than one vehicle. This procedure allows generating less number of routes. We repeat this procedure until the capacity or time- window constraint is violated or all the customers are assigned to a vehicle.

In this step, the time balance of the routes generated are also considered. We define the parameters: total service time (TST), single route length (SRL), minimum number of vehicles used (N), and the minimum travel time from depot (MD). TST is the sum of service time of all customers. SRL is calculated using NN algorithm [Bräysy and Gendreau, 2005], so that it will give the total travel time of a single route that contains all the customers. N is found by dividing total demand of customers to the capacity of the largest vehicle. Finally, MD is the sum of the travel times from the depot of the closest N customers. We calculate an average total time (ATT) of a route using the following formula:

$$ATT = \frac{TST + SRL + MD}{N}.$$

We also define a balance fraction (BF) parameter, which is between 0 and 1. A route is constructed if the total route length is between $(ATT - ATT * BF)$ and $(ATT + ATT * BF)$.

As we discuss in Section 3.2, choosing between NN and priority assignment depends on the size and type of the data. For some data sets NN gives better results, whereas priority assignment overcomes NN for others. Therefore, the algorithm initializes the diverse solutions by using both NN algorithm and priority assignment. The set of solutions prepared by both methods are compared and the superior set is chosen and continue to the next steps of the algorithm.

Improvement Methods

After the initial solutions are generated, a sequence of improvement methods are applied. These methods are also based on the work of Belfiore and Yoshizaki [Belfiore and Yoshizaki, 2013]. The improvement steps are: swap of customers in the same route, demand reallocation of customers, elimination and combination of the routes, insertion and route addition. These improvement methods are useful for decreasing total time spent in a route and decreasing the number of vehicles used. Swap, elimination and combination of the routes and insertion steps are the same as in Section 3.2.

Demand Reallocation

In this procedure, we count the number of split customers. Among the split customers, we select the customer i that is farthest from the depot. Then, we select the routes that contain customer i. Let R_k be the route that contains the customer i. We select another route R_t that also contains customer i. If R_t has enough empty capacity to reallocate the demand of the customer in R_k, then customer i will be eliminated from route R_k and the demand that is served in R_k will be reallocated to route R_t. We allow this procedure only if the total time or total number of vehicles are reduced. The procedure is repeated for each split customer and for each route that contains the split customer.

Sometimes, a route contains only one customer which is split into more than one route. In that case, reallocation step is applied to reduce the total number of vehicles.

Also, a route that contains split customer may have a long travel time. To reduce the total travel time, demand reallocation procedure is useful.

Route Addition

After route elimination and combination, and insertion steps, if there are remaining customers that are split, route addition procedure is applied. First, we determine the number of split customers. Then, among them we start the procedure with the customer i which is farthest from the depot. We select a route R_k among the route set that contains customer i. The procedure continues with the selection of a new route, R_t which is different from R_k, from the whole route list. If there is enough capacity left for the demand of customer i in the route R_t while the time windows are not violated and the total time decreases, then customer i is added to the route R_t.

The addition procedure continues for each split customer and for each route. The candidate routes are generated for each customer. The solution that gives the minimum total travel time and minimum number of vehicles will be selected.

After the improvement steps are applied, the algorithm continues with reference set update and combination methods, which are also discussed in Section 3.2. The algorithm continues until maximum number of allowed iterations, $IterNo$, is reached. In the final step, there are $IterNo$ number of solutions generated in each iteration. The best solution, that has the minimum total travel time and minimum number of vehicles, is selected among the candidate solutions. We test our algorithm with different $PSize$, b_1, b_2 and $IterNo$ values. For different data sets, the parameters can take different values. A detailed explanation is given in Chapter 4.

Chapter 4

COMPUTATIONAL RESULTS

In this chapter, we compare the results gained from the mathematical and heuristic models that we develop in Chapter 3. We describe the data and the computational environment in which we test our models, in sections 4.1 and 4.2. Section 4.3 is dedicated to the results of VRPTWRB and SDVRPTWRB. In the last section, we discuss about how we choose parameters.

4.1 Test Problems

The data used are collected from three companies, a manufacturing company, a logistics company and a FMCG company, based in Turkey. While in the first and second data sets, the customers are located in İstanbul, the third data set consists of customers spread over Turkey. In each data set, customers have time windows and service times. The fleet of vehicles varies with capacities and fixed costs. A detailed description of the data is available in Appendix A. We use Google Distance Matrix API to get the travel time and distance matrix between the customers and the depot, Google API considers the traffic congestion at any time, thus we use real time travel data.

We also compare our heuristic algorithm with Solomon benchmark problems, R1, R2, C1, C2, RC1, RC2, with 100 customers [Solomon, 2008]. There are three problem sets in terms of customer locations. R1 and R2 problems contain randomly generated geographical data of customers, C1 and C2 problems contain clustered customers and in RC1 and RC2 problems some customers are randomly located and some are clustered. In terms of planning horizon, the problem sets are divided in two: R1, C1 and RC1 problems have very short planning horizon thus only a few customers are contained in a route. Whereas R2, C2 and RC2 problems have long scheduling horizon, so more than 30 customers can be on the same route. These problems are euclidean problems in which travel times between the customers are equal to the corresponding distances.

4.2 Computational Platform

We use IBM ILOG CPLEX Optimization Studio version 12.5.1 to execute the exact model of VRPTWRB on a server equipped with Intel (R) Xeon(R) CPU E5-1660 v3 @ 3.00 GHz processor and 32 GB RAM. The absolute MIP gap tolerance is 0.000001 by default and we set time limit to 24 hours. The execution terminates when gap tolerance or time limit is reached.

The heuristic algorithm is executed on Java on a computer equipped with Intel(R) Core(TM) i7-6600U CPU @ 2.60GHz processor and 8 GB RAM.

4.3 Computational Results

4.3.1 Results of VRPTWRB

In order to evaluate the performance of our VRPTWRB exact model in Section 3.1, we compare it with a VRP model that both considers time windows and route balance[Sivaramkumar et al., 2018]. In the benchmark problem, Sivaramkumar et al. consider route balance as total time balance. Therefore, the objective is to minimize the difference between the longest and the shortest routes in terms of total time. Their mathematical model consists of capacity, time-window and maximum route time constraints. The difference between the benchmark model and our model is the route balance constraints. They only consider route balance with computing the travel time of the longest and the shortest routes. However, the benchmark model is the closest mathematical model that can be compared in the literature.

Table 4.1 shows the comparison of the total travel time, total number of vehicles and CPU time of the exact model of Sivaramkumar et al., our exact model and SS algorithm.The data collected from the companies in Turkey is tested. The data sets are described with the number of customers and number of available vehicles. For example, in the first row, C13V27 means there are 13 customers and 27 available vehicles in data set 1. For the exact models, % gap information is also given. We can see from the table that the total time and number of vehicles increases in our exact model. This is an expected result since we only consider the total time as objective. Unlike existing approaches, our models create balance among the routes by balancing the departure and arrival times of each vehicle. To compare how balanced the routes generated, we compute the mean and standard deviation of each solution. The comparison of mean and standard deviation of each solution is given in Appendix B, Table B.1. Thus, it is expected to get higher waiting times, which affects the objective. According to the data, our SS algorithm gives better results compared to existing exact model in the literature. The total travel times are increased but the number of vehicles used are decreased with our algorithm.

In terms of CPU times, our SS algorithm works significantly faster than the exact models for large size problems. Mostly in less than a minute, the algorithm gives promising results.

In chapter 3, we state that our scatter search based heuristic algorithm is superior than the mathematical model since dynamic traffic time and distance values are considered. In Table 4.1, we compare the heuristic algorithm with the exact models by using a prfe-defined distance and time matrix. Therefore, the dynamic values are not used. However, in Table 4.2 we give the heuristics results using dynamic traffic data. We can conclude that the scatter search based heuristic algorithm is superior than the exact models in terms of generating real time routes.

Table 4.1: Comparison Table of Exact Models and Scatter Search Algorithm (in terms of total time)

	Sivaramkumar et al. Exact Model				**VRPTWRB Exact Model**				**Scatter Search Algorithm**		
Data set	**Total time (min)**	**Total number of vehicles**	**CPU time (sec)**	**% Gap**	**Total time (min)**	**Total number of vehicles**	**CPU time (sec)**	**% Gap**	**Total time (min)**	**Total number of vehicles**	**CPU time (sec)**
1(C13V27)	1021	6	14	0%	1021	6	12	0%	1116	4	12
2(C19V27)	1473	10	14	0%	1473	10	51	0%	1587	11	65
3(C11V25)	501.9	4	8700	0%	709.433	6	4	0%	758	5	58
4(C5V25)	282.45	3	1	0%	282.45	3	1	0%	319.817	2	1
5(C12V25)	453.417	4	16	0%	618.333	5	304	0%	736.56	4	2
6(C12V25)	587.867	4	14	0%	830.333	6	26	0%	864.32	5	2
7(C14V25)	234.967	6	26	0%	253.533	7	36	0%	425.69	5	1
8(C22V25)	931.983	8	86400*	33.39%	1117.23	10	86400*	23.58%	1351.13	8	80
9(C17V25)	577.183	6	86400*	10.85%	825.417	9	86400*	4.34%	875.15	6	54
10(C11V25)	692.9	5	16	0%	831.067	6	7	0%	898.57	5	6
11(C22V25)	1444.1	9	86400*	33.56%	668.5	8	86400*	5.93%	1813.12	8	83
12(C14V25)	694.6	11	86400*	58.70%	784.083	7	86400*	4.91%	839.96	5	5
13(C14V25)	747.85	5	86400*	48.30%	892.35	6	4080	0%	1113.25	5	1
14(C16V25)	817.817	6	86400*	49%	840.4	7	4320	0%	1010.13	6	4

* Time limit reached.

Table 4.2: Comparison Table of Exact Models and Scatter Search Algorithm with dynamic traffic data (in terms of total time)

	Sivaramkumar et al. Exact Model				**VRPTWRB Exact Model**				**Scatter Search Algorithm**		
Data set	**Total time (min)**	**Total number of vehicles**	**CPU time (sec)**	**% Gap**	**Total time (min)**	**Total number of vehicles**	**CPU time (sec)**	**% Gap**	**Total time (min)**	**Total number of vehicles**	**CPU time (sec)**
1(C13V27)	1021	6	14	0%	1021	6	**12**	0%	1094	**4**	16
2(C19V27)	1473	10	14	0%	1473	10	51	0%	**1147**	11	68
3(C11V25)	501.90	4	8700	0%	709.43	6	4	0%	755	**5**	**65**
4(C5V25)	282.45	3	1	0%	282.45	3	1	0%	319.82	**2**	**1**
5(C12V25)	453.42	4	16	0%	618.33	5	304	0%	719.70	**4**	**2**
6(C12V25)	587.87	4	14	0%	830.33	6	26	0%	862.86	5	**3**
7(C14V25)	234.97	6	26	0%	253.53	7	36	0%	425.46	**5**	**4**
8(C22V25)	931.98	8	86400*	33.39%	1117.23	10	86400*	23.58%	1341.27	**8**	**71**
9(C17V25)	577.18	6	86400*	10.85%	825.42	9	86400*	4.34%	888.24	6	**62**
10(C11V25)	692.90	5	16	0%	831.07	6	7	0%	883.39	**5**	**4**
11(C22V25)	1444.10	9	86400*	33.56%	668.50	8	86400*	5.93%	1678.05	**8**	**70**
12(C14V25)	694.60	11	86400*	58.70%	784.08	7	86400*	4.91%	824.81	**5**	**2**
13(C14V25)	747.85	5	86400*	48.30%	892.35	6	4080	0%	986.37	**5**	**2**
14(C16V25)	817.82	6	86400*	49%	840.40	7	4320	0%	986.66	**6**	**4**

* Time limit reached.

Table 4.3 compares the same models with the same data set, but with another objective: the difference between longest and shortest routes generated. With this objective, we can see that our models give better results than the model in the literature compared to the total travel time objective. The VRPTWRB model obtains comparable results with Sivaramkumar's exact model in terms of total number of vehicles used. It is seen in Table 4.3 that, if the number of vehicles found by VRPTWRB model are less than the compared model, the travel time difference between the longest and shortest routes of VRPTWRB is larger than the compared model. If the number of vehicles are the same with the compared model, then this difference is less then the compared model. In Appendix B, in Table B.2, the average times of the routes generated are higher than the results of the benchmark model. However, the deviation between the routes are much less than the results of the benchmark model. Therefore, we can generate better balanced routes. Thus, we can have better results with VRPTWRB. For the SS algorithm, the results are mostly comparable with the exact results. In addition, SS algorithm works significantly faster than the exact models.

Similar to Table 4.2, in Table 4.4 we compare the scatter search based heuristic algorithm using dynamic traffic values with the exact models. We also conclude from this table that the heuristic algorithm is superior than the exact models when we use dynamic traffic data.

Table 4.3: Comparison Table of Exact Models and Scatter Search Algorithm (in terms of L_{max} - L_{min})

	Sivaramkumar et al. Exact Model				**VRPTWRB Exact Model**				**Scatter Search Algorithm**		
Data set	**Obj. (min)**	**Total number of vehicles**	**CPU time (sec)**	**% Gap**	**Obj. (min)**	**Total number of vehicles**	**CPU time (sec)**	**% Gap**	**Obj. (min)**	**Total number of vehicles**	**CPU time (sec)**
1(C13V27)	339	6	600	0%	48	8	16	0%	374	7	6
2(C19V27)	258	17	86400*	3.17%	396	13	5	0%	408	13	4
3(C11V25)	115	11	18	0%	243	7	3	0%	261.88	8	1
4(C20V25)	22	20	3645	0%	6	14	31	0%	46	15	3
5(C5V25)	138	6	2	0%	0	3	1	0%	142.69	3	1
6(C12V25)	125	12	29	0%	16.717	8	61	0%	36.71	9	3
7(C12V25)	1417	6	5340	0%	238	6	2	0%	308.1	6	2
8(C14V25)	14	10	86400*	54.59%	19.083	8	27	0%	23.93	9	4
9(C22V25)	13	20	456	0%	275.3	10	86400*	11.01%	298.37	11	3
10(C17V25)	9.867	10	10140	0%	237	10	6	0%	306.25	12	3
11(C11V25)	18	10	3	0%	0	8	2	0%	37.11	10	2
12(C14V25)	103.25	13	141	0%	19.683	8	17	0%	30.98	9	4
13(C14V25)	150	13	7226	0%	24.483	8	23	0%	43.58	8	5
14(C16V25)	123	15	117	0%	0	12	49	0%	99.03	13	2
15(C22V25)	36	15	8	0%	251	13	458	0%	394.95	14	6
16(C22V25)	7.833	20	35	0%	242	12	64	0%	267.73	13	6
17(C28V25)	68	27	86400*	19.60%	237	16	226	0%	284.6	18	7
18(C34V25)	45	29	86400*	70.66%	243	22	4680	0%	283.58	25	11

* Time limit reached.

Table 4.4: Comparison Table of Exact Models and Scatter Search Algorithm with dynamic traffic data (in terms of L_{max} - L_{min})

	Sivaramkumar et al. Exact Model				**VRPTWRB Exact Model**				**Scatter Search Algorithm**		
Data set	**Obj. (min)**	**Total number of vehicles**	**CPU time (sec)**	**% Gap**	**Obj. (min)**	**Total number of vehicles**	**CPU time (sec)**	**% Gap**	**Obj. (min)**	**Total number of vehicles**	**CPU time (sec)**
1(C13V27)	339	6	600	0%	**48**	8	16	0%	**193**	7	10
2(C19V27)	258	17	86400*	3.17%	396	**13**	5	0%	396	**13**	16
3(C11V25)	115	11	18	0.00%	243	**7**	3	0%	**41.683**	**8**	8
4(C20V25)	22	20	3645	0%	**6**	**14**	31	0%	47.7	**15**	42
5(C5V25)	138	6	2	0%	**0**	**3**	1	0%	138.2	**3**	1
6(C12V25)	125	12	29	0%	**16.717**	**8**	61	0%	**43.32**	**9**	12
7(C12V25)	1417	6	5340	0%	**238**	6	2	0%	**243.57**	6	9
8(C14V25)	14	10	86400*	54.59%	19.08	**8**	27	0%	49.78	**9**	11
9(C22V25)	13	20	456	0%	275.3	**10**	86400*	11.01%	127.82	**11**	26
10(C17V25)	9.867	10	10140	0%	237	10	**6**	0%	125.27	12	18
11(C11V25)	18	10	3	0%	**12**	**8**	2	0%	35.67	10	6
12(C14V25)	103.25	13	141	0%	**19.68**	**8**	17	0%	**50.683**	**9**	21
13(C14V25)	150	13	7226	0%	**24.48**	**8**	23	0%	**37.75**	**8**	19
14(C16V25)	123	15	117	0%	**23**	**12**	49	0%	**39.25**	**13**	24
15(C22V25)	36	15	8	0%	251	**13**	458	0%	93.95	**14**	56
16(C22V25)	7.833	20	35	0%	242	**12**	64	0%	116.45	**13**	47
17(C28V25)	68	27	86400*	19.60%	237	**16**	226	0%	**54.35**	**18**	98
18(C34V25)	45	29	86400*	70.66%	243	**22**	4680	0%	67.05	**25**	143

* Time limit reached.

For Solomon benchmark problems, we compare our SS algorithm results with the best possible results found with heuristic models in the literature (`http://web.cba.neu.edu/~msolomon/problems.htm`). Tables 4.5, 4.6, 4.7, 4.8, 4.9, 4.10 show the results of R1, C1, RC1, R2, C2 and RC2, respectively. For most of the problem sets, our algorithm gives better results, in terms of total distance and total number of vehicles. We also calculate the mean and standard deviation of the routes generated for each problem set. For the clustered problem set with short scheduling horizon, C1, the differences of travel times between the routes are minimum among the benchmark problem sets. For these problems, the number of vehicles used do not change but the total distance decreases and balance between the routes are satisfied.

For R1 and RC1 problem sets, total distance and the number of vehicles used decrease. However, in terms of route balance the results are not sufficient. Since the number of vehicles used in R2, C2 and RC2 problem sets is small, it becomes harder to create a balance between the routes. Thus, the standard deviations are large in these problem sets. For R2, total distance increases on the average. However, we can generate routes with less number of vehicles. Therefore, the increase in the total distance is reasonable.

Table 4.5: Results of Solomon R1 Problems with 100 customers

	Best Possible Result		**SS Algorithm**			
Problem set	**Total number of vehicles**	**Total distance**	**Total number of vehicles**	**Total distance**	**Mean (μ)**	**Standard deviation (σ)**
R101	19	1645.79	17	1004.37	59.08	17.48
R102	17	1486.12	14	1003.97	71.71	30.30
R103	13	1292.68	12	1003.34	83.61	40.51
R104	9	1007.24	12	1003.34	83.61	40.51
R105	14	1377.11	11	1003.32	91.21	29.44
R106	12	1251.98	10	1003.28	100.33	38.01
R107	10	1104.66	10	1002.965	100.30	37.20
R108	9	960.88	10	1002.61	100.26	36.38
R109	11	1194.73	10	1002.95	100.30	38.00
R110	10	1118.59	10	1003.23	100.32	37.47
R111	10	1096.72	9	1003.07	111.45	46.76
R112	9	982.14	8	1002.52	125.32	27.89
Average % of Improvements			**5 %**	**15 %**		

Table 4.6: Results of Solomon C1 Problems with 100 customers

	Best Possible Result		**SS Algorithm**			
Problem set	**Total number of vehicles**	**Total distance**	**Total number of vehicles**	**Total distance**	**Mean (μ)**	**Standard deviation (σ)**
C101	10	828.94	10	603.613	60.36	0.13
C102	10	828.94	10	603.54	60.35	0.16
C103	10	828.06	10	603.471	60.35	0.14
C104	10	824.78	10	603.726	60.37	0.17
C105	10	828.94	10	603.694	60.37	0.15
C106	10	828.94	10	603.483	60.35	0.17
C107	10	828.94	10	603.478	60.35	0.17
C108	10	828.94	10	603.619	60.36	0.15
Average % of Improvements			**0 %**	**27 %**		

Table 4.7: Results of Solomon RC1 Problems with 100 customers

	Best Possible Result		**SS Algorithm**			
Problem set	**Total number of vehicles**	**Total distance**	**Total number of vehicles**	**Total distance**	**Mean (μ)**	**Standard deviation (σ)**
RC101	14	1696.94	12	1003.516	83.63	23.32
RC102	12	1554.75	11	1003.204	91.20	36.68
RC103	11	1261.67	11	1003.12	91.19	40.59
RC104	10	1135.48	10	1002.834	100.28	45.34
RC105	13	1629.44	12	1003.923	83.66	37.22
RC106	11	1424.73	11	1003.165	91.20	34.09
RC107	11	1230.48	11	1003.181	91.20	31.00
RC108	10	1139.82	10	1002.694	100.27	35.88
Average % of Improvements			**4 %**	**26 %**		

Table 4.8: Results of Solomon R2 Problems with 100 customers

	Best Possible Result		**SS Algorithm**			
Problem set	**Total number of vehicles**	**Total distance**	**Total number of vehicles**	**Total distance**	**Mean (μ)**	**Standard deviation (σ)**
R201	4	1252.37	2	1245.03	622.52	9.22
R202	3	1191.7	2	1241.2	620.60	59.71
R203	3	939.54	2	1242.25	621.13	23.30
R204	2	825.52	2	859.75	429.88	7.22
R205	3	994.42	2	764.99	382.50	7.10
R206	3	906.14	2	864.98	432.49	22.70
R207	2	893.33	2	934.45	467.23	113.22
R208	2	726.75	2	861.9	430.95	169.35
R209	3	909.16	2	934.2	467.10	162.01
R210	3	939.34	2	904.49	452.25	97.04
R211	2	892.71	2	864.48	432.24	181.20
Average % of Improvements			**23 %**	**-3 %**		

Table 4.9: Results of Solomon C2 Problems with 100 customers

	Best Possible Result		**SS Algorithm**			
Problem set	**Total number of vehicles**	**Total distance**	**Total number of vehicles**	**Total distance**	**Mean (μ)**	**Standard deviation (σ)**
C201	3	591.56	3	455.7	151.90	37.43
C202	3	591.56	3	441.6	147.20	42.41
C203	3	591.17	3	430.2	143.40	38.03
C204	3	590.6	3	431.7	143.90	41.65
C205	3	588.88	3	440.2	146.73	36.42
C206	3	588.49	3	441.7	147.23	33.00
C207	3	588.29	3	437.6	145.87	30.70
C208	3	588.32	3	438.9	146.30	25.11
Average % of Improvements			**0 %**	**25 %**		

Table 4.10: Results of Solomon RC2 Problems with 100 customers

	Best Possible Result		**SS Algorithm**			
Problem set	**Total number of vehicles**	**Total distance**	**Total number of vehicles**	**Total distance**	**Mean (μ)**	**Standard deviation (σ)**
RC201	4	1406.91	2	1243.9	621.95	3.87
RC202	3	1367.09	2	1242.7	621.35	3.05
RC203	3	1049.62	2	1247.37	623.69	5.13
RC204	3	798.41	2	694.3	347.15	147.83
RC205	4	1297.19	2	1242.99	621.50	4.85
RC206	3	1146.32	2	1146.09	573.05	52.47
RC207	3	1061.14	2	1006.25	503.13	115.47
RC208	3	828.14	2	1006.19	503.10	115.94
Average % of Improvements			**38 %**	**0 %**		

4.3.2 Results of SDVRPTWRB

For SDVRPTWRB, we validate the exact model that is developed in Section 3.3.1 by comparing with a mathematical model developed for solving VRP with time windows and split delivery in the literature [Belfiore and Yoshizaki, 2013]. The benchmark model considers only split delivery with time-windows. Their mathematical model consists of capacity and time-window restrictions and the constraints that allow split delivery, while our model also considers route balance. We also test our SS with split delivery algorithm.

In order to evaluate the performance of our exact model, we compare it with a VRP model that both considers time windows and scatter search[Belfiore and Yoshizaki, 2013].

Table 4.11 shows the comparison of the total travel time, total number of vehicles and CPU time of the exact model in the literature and our exact model. The data collected from the companies in Turkey is tested. The data sets are described with the number of customers and number of available vehicles. We conclude that the compared exact model gives slightly better results in terms of total travel time when the models are optimally solved. This result is expected since Belfiore and Yoshizaki do not consider route balance in their model. Thus, their model can give solutions with less number of vehicle or total travel time. Our model, on the other hand, both considers route balance and permits scatter search. The mean and standard deviation of the routes generated for each solution is shown in Appendix B, Table B.3. Also, our model is validated to perform well since the results are reasonable and comparable with the model of Belfiore and Yoshizaki.

Table 4.11: Comparison Table of Exact Models and Scatter Search Algorithm (in terms of total time)

	Belfiore and Yoshizaki Exact Model				SDVRPTWRB Exact Model			
Data set	**Total time (min)**	**Total number of vehicles**	**CPU time (sec)**	**% Gap**	**Total time (min)**	**Total number of vehicles**	**CPU time (sec)**	**% Gap**
1(C13V27)	2474	7	3	0%	2474	7	6	0%
2(C19V27)	2278	7	7	0%	2552	9	47	0%
3(C11V25)	2089.17	6	1	0%	2089.89	6	2	0%
4(C5V25)	456	5	1	0%	456	5	1	0%
5(C12V25)	109.25	1	86400	18.50%	269.42	5	59	0%
6(C12V25)	800.45	5	1	0%	811.6	6	1	0%
7(C14V25)	235.75	7	1	0%	261.47	8	3	0%
9(C17V25)	877.05	9	2	0%	880.64	10	8	0%
10(C11V25)	1313.70	6	1	0%	1320.19	7	4	0%
11(C22V25)	1569.69	10	5	0%	1569.69	10	660	0%
12(C14V25)	1727.15	5	1	0%	1766.51	8	61	0%
13(C14V25)	1389.82	5	4	0%	1403.64	7	7	0%
14(C16V25)	1092.73	4	3	0%	1133.73	8	27	0%
18(C20V25)	377.39	10	2	0%	391.47	10	9	0%

For Solomon benchmark problems, we compare our SS algorithm results with the best possible results found with heuristic models in the literature [Belfiore and Yoshizaki, 2013]. Tables 4.12, 4.13, 4.14, 4.15, 4.16, 4.17 show the results of R1, C1, RC1, R2, C2 and RC2, respectively. For most of the problem sets, our algorithm gives better results, in terms of total distance and total number of vehicles. We also calculate the mean and standard deviation of the routes generated for each problem set. For the clustered problem set with short scheduling horizon, C1, the differences of travel times between the routes are minimum among the benchmark problem sets. For these problems, the number of vehicles used do not change but the total distance decreases and balance between the routes are satisfied.

For RC1 and C2 problem sets, our algorithm also gives reasonable results. For RC1, the number of vehicles used is high for each instance. Thus, the deviation is reasonable. For example, for RC106, there are 12 routes generated and the deviation of the routes are 18 minutes. Thus, a route balance is satisfied among the routes.

For R1 and R2 problem sets, overall, total distance and the number of vehicles used decrease. However, in terms of route balance the results are not sufficient. Since the customers are randomly located in these problem sets, it becomes harder to create a balance between the routes.

For RC2 problem set, we can see that our algorithm performs as good as the algorithm of Belfiore and Yoshizaki. The route balance between the routes is also satisfied, except the instances RC204 and RC208. For some of the instances, the total distance increases in our solutions.

Table 4.12: SDVRPTWRB Results of Solomon R1 Problems with 100 customers

	Belfiore and Yoshizaki		**SS Algorithm**			
Problem set	**Total number of vehicles**	**Total distance**	**Total number of vehicles**	**Total distance**	**Mean (μ)**	**Standard deviation (σ)**
R101	19	1651.84	**17**	**1054.19**	62.01	15.84
R102	17	1495.83	**16**	**1104.42**	69.03	33.21
R103	14	1302.41	**13**	**1103.68**	84.90	41.85
R104	10	989.32	13	1103.68	84.90	41.85
R105	14	1450.27	**12**	**1103.76**	91.98	34.50
R106	12	1283.75	**11**	**1053.28**	95.75	46.07
R107	10	1134.96	10	**1102.89**	110.29	35.95
R108	9	993.58	10	1102.97	110.30	35.10
R109	12	1161.93	**10**	**1103.08**	110.31	35.12
R110	10	1172.47	10	**1103.41**	110.34	36.51
R111	11	1076.48	**10**	1103.11	110.31	43.76
R112	10	970.06	**8**	1102.79	137.85	25.51
Average % of Improvements			**4 %**	**8 %**		

Table 4.13: SDVRPTWRB Results of Solomon C1 Problems with 100 customers

	Belfiore and Yoshizaki		**SS Algorithm**			
Problem set	**Total number of vehicles**	**Total distance**	**Total number of vehicles**	**Total distance**	**Mean (μ)**	**Standard deviation (σ)**
C101	10	828.94	10	**604.46**	60.45	0.12
C102	10	838.90	10	**604.25**	60.43	0.12
C103	10	832.48	10	**604.36**	60.44	0.12
C104	10	840.65	10	**604.19**	60.42	0.12
C105	10	841.34	10	**603.06**	60.31	0.12
C106	10	839.63	10	**603.86**	60.39	0.12
C107	10	828.94	10	**604.15**	60.42	0.11
C108	10	828.94	10	**604.05**	60.41	0.12
Average % of Improvements			**0 %**	**28 %**		

Table 4.14: SDVRPTWRB Results of Solomon RC1 Problems with 100 customers

	Belfiore and Yoshizaki		**SS Algorithm**			
Problem set	**Total number of vehicles**	**Total distance**	**Total number of vehicles**	**Total distance**	**Mean (μ)**	**Standard deviation (σ)**
RC101	15	1640.39	**13**	**1143.95**	88.00	23.32
RC102	13	1491.12	**11**	**1143.96**	104.00	36.68
RC103	11	1307.88	11	**1143.66**	103.97	39.38
RC104	10	1170.81	11	**1143.13**	103.92	29.34
RC105	13	1678.41	**12**	**1003.93**	83.66	34.55
RC106	12	1409.81	12	1504.29	125.36	18.01
RC107	11	1268.34	11	**1003.18**	91.20	36.94
RC108	10	1139.82	10	**1002.69**	100.27	35.31
Average % of Improvements			**3 %**	**17 %**		

Table 4.15: SDVRPTWRB Results of Solomon R2 Problems with 100 customers

	Belfiore and Yoshizaki		**SS Algorithm**			
Problem set	**Total number of vehicles**	**Total distance**	**Total number of vehicles**	**Total distance**	**Mean (μ)**	**Standard deviation (σ)**
R201	4	1283.41	**3**	1471.4	490.47	49.39
R202	3	1237.12	3	1443.51	481.17	6.16
R203	3	980.62	3	1303.38	434.46	5.45
R204	2	855.79	2	935.9	467.95	61.10
R205	3	1045.69	**2**	**977.82**	488.91	113.95
R206	3	958.24	**2**	**834.5**	417.25	197.69
R207	2	937.40	2	**933.42**	466.71	164.91
R208	2	756.99	2	867.96	433.98	115.45
R209	3	954.77	2	**934.32**	467.16	146.98
R210	3	971.73	2	**933.19**	466.60	148.75
R211	3	833.45	2	1004.2	502.10	128.01
Average % of Improvements			**17 %**	**-8 %**		

Table 4.16: SDVRPTWRB Results of Solomon C2 Problems with 100 customers

	Belfiore and Yoshizaki		SS Algorithm			
Problem set	**Total number of vehicles**	**Total distance**	**Total number of vehicles**	**Total distance**	**Mean (μ)**	**Standard deviation (σ)**
C201	3	599.37	3	**432.4**	144.13	26.16
C202	3	591.56	4	**386.4**	96.60	22.46
C203	3	591.17	3	**415.5**	138.50	11.90
C204	3	614.55	4	**419.6**	104.90	25.68
C205	3	599.08	3	**432.5**	144.17	28.22
C206	3	597.32	3	**418.9**	139.63	32.59
C207	3	588.29	3	**403.8**	134.60	18.00
C208	3	588.32	3	**433.7**	144.57	21.48
Average % of Improvements			**3 %**	**30 %**		

Table 4.17: SDVRPTWRB Results of Solomon RC2 Problems with 100 customers

	Belfiore and Yoshizaki		SS Algorithm			
Problem set	**Total number of vehicles**	**Total distance**	**Total number of vehicles**	**Total distance**	**Mean (μ)**	**Standard deviation (σ)**
RC201	4	1454.62	**3**	1509.41	503.14	11.56
RC202	3	1370.71	3	**1010.03**	336.68	14.51
RC203	3	1053.96	3	1250.18	416.73	4.18
RC204	3	833.89	3	**606.3**	202.10	117.75
RC205	4	1337.81	3	1368.96	456.32	11.49
RC206	4	1127.44	**2**	**1008.54**	504.27	10.57
RC207	4	1045.55	**2**	1145.63	572.81	18.70
RC208	3	886.75	**2**	1002.23	501.12	116.64
Average % of Improvements			**23 %**	**2 %**		

4.4 Parameter Selection

Recall that in the heuristic algorithms we have some parameters, such as $PSize$, b_1, b_2 and $IterNo$. These parameters may vary due to the characterization of the data sets. We test each problem set with different parameter combinations, which is shown in Table 4.18. These parameter combinations are taken from the literature [Belfiore and Yoshizaki, 2013]. Each row under $PSize$, b_1, b_2 columns represents the parameter combination used. We run the algorithm with these combinations for each iteration represented under $IterNo$ column. For example, we run the algorithm with $PSize = 5$, $b_1 = 3$, $b_2 = 2$ with iterations 1, 2, 5, 10 and 15.

Using these parameter combinations, we run each instance. For the data sets that show the same characteristics, the parameter combinations that give the best results for VRPTWRB and SDVRPTWRB are shown in Table 4.19 and 4.20, respectively.

Table 4.18: Parameter Combinations

$PSize$	b_1	b_2	$IterNo$
5	3	2	1
10	6	4	2
15	10	5	5
20	15	5	10
25	15	10	15
30	25	5	
30	20	10	

Table 4.19: Best Parameter Combinations for VRPTWRB

DataSet	*PSize*	b_1	b_2	*IterNo*
Solomon R1	30	25	5	1
Solomon C1	10	6	4	1
Solomon RC1	30	25	5	1
Solomon R2	20	15	5	2
Solomon C2	25	15	10	5
Solomon RC2	15	10	5	15
Data 1-2	30	25	5	2
Data 3-18	5	3	2	1

Table 4.20: Best Parameter Combinations for SDVRPTWRB

DataSet	*PSize*	b_1	b_2	*IterNo*
Solomon R1	20	15	5	1
Solomon C1	30	20	10	15
Solomon RC1	20	15	5	1
Solomon R2	10	6	4	1
Solomon C2	15	10	5	5
Solomon RC2	15	10	5	1
Data 1-2	15	10	5	2

Chapter 5

CONCLUSIONS

The vehicle routing problem is a well- known problem in operations research with many applications. In this thesis, we consider the vehicle routing problem with time windows and route balance. We define route balance as balancing the departure and arrival times of the routes, which is a different approach from the literature. We also extend this problem to split delivery. Two mathematical models and a scatter search based heuristic algorithm that works for both problems are developed.

The objectives of the mathematical models are to minimize total travel time or the difference between the longest and the shortest routes. We compare our exact models with the mathematical models that are developed in the literature and conclude that our models create a better balance between the routes generated. In VRPTWRB, our exact model reduces the total number of vehicles used by 22%. We also compare our algorithm and conclude that for most of the problem instances, it gives reasonable solutions. Our heuristic algorithm reduces the total number of vehicles used by 18 %.

In developing the scatter search-based heuristic algorithm, our aim is to find near optimal solutions in a short amount of time. For this purpose, we test our algorithm with Solomon benchmark problems, and with the data collected from three different companies (manufacturing, FMCG and logistics) in Turkey. All of the results are generated in less than five minutes. Thus, our algorithm performs well and fast. For Solomon benchmark problems, in VRPTWRB the heuristic algorithm reduces the number of vehicles used and total distance by 12% and 15%, respectively. In SDVRPTWRB, the improvements in the total number of vehicles used and total distance traveled are 9% and 13%, respectively.

The balance among the routes is becoming an important issue; equity among the workers is essential. Thus, new models and approaches are needed to create effective and efficient routes.

5.1 Future Work

In our exact models, different objectives rather than total travel time or the difference between the longest and shortest routes can be implemented. In addition, weighted combination of total travel time and the difference between the longest and shortest routes can be studied as an objective function. Multiple commodity and multiple depot extensions of the model can also be considered.

In the scatter search based algorithm, different constructive heuristics other than the ones stated in the thesis can be used. In the improvement steps, different heuristic methods can be used. For creating route balance, a fixed balance percent or different calculations may be implemented.

In our mathematical model we are given the deterministic travel time and distance matrices. As a future work, the optimization model can be solved by using stochastic travel times and distances.

We develop our models only in an economic perspective. We consider route balance as a social issue, however the constraints and the objectives are in terms of travel times. New constraints and objectives can be implemented to the model which are constructed on motivation of workers and drivers. For example, maximizing the drivers' motivation and quality of working conditions can be an objective.

The environmental perspective is also becoming an important issue. Numerous studies considering green VRPs state that the vehicles used in routing cause air pollution and green house gas (GHG) emissions. A model can be developed to minimize CO_2 and GHG emissions of the vehicles. Currently, there are proposed models that use related objectives and constraints in the literature. A route balance extension can be implemented to these models.

A better model is to be the one having environmental, social and economical perspective, simultaneously. The objective of the model can be a combination of these three perspectives. The companies start being interested in the models with the triple- bottom line objectives. Therefore, in the near future, there will be more of the applications of these models.

BIBLIOGRAPHY

[Alegre et al., 2007] Alegre, J., Laguna, M., and Pacheco, J. (2007). Optimizing the periodic pick-up of raw materials for a manufacturer of auto parts. *European Journal of Operational Research*.

[Archetti and Speranza, 2012] Archetti, C. and Speranza, M. G. (2012). Vehicle routing problems with split deliveries. *International Transactions in Operational Research*, 19(1-2):3–22.

[Archetti et al., 2006] Archetti, C., Speranza, M. G., and Hertz, A. (2006). A Tabu Search Algorithm for the Split Delivery Vehicle Routing Problem. *Transportation Science*.

[Baker and Schaffer, 1986] Baker, E. K. and Schaffer, J. R. (1986). Solution Improvement Heuristics for the Vehicle Routing and Scheduling Problem with Time Window Constraints. *American Journal of Mathematical and Management Sciences*, 6(3-4):261–300.

[Balinski and Quandt, 1964] Balinski, M. L. and Quandt, R. E. (1964). On an Integer Program for a Delivery Problem. *Operations Research*, 12(2):300–304.

[Belfiore and Yoshizaki, 2013] Belfiore, P. and Yoshizaki, H. T. (2013). Heuristic methods for the fleet size and mix vehicle routing problem with time windows and split deliveries. *Computers & Industrial Engineering*, 64(2):589–601.

[Bräysy and Gendreau, 2005] Bräysy, O. and Gendreau, M. (2005). Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms. *Transportation Science*.

[Burrows et al., 1972] Burrows, C., Eilon, S., Watson-Gandy, C., and Christofides, N. (1972). Distribution management: Mathematical Modelling and Practical Analysis. *Applied Statistics*, 21:337.

[Chen and Chen, 2008] Chen, J. and Chen, S. (2008). Optimization of Vehicle Routing Problem with Load Balancing and Time Windows in Distribution. In *IEEE Xplore*. IEEE.

[Chiang and Russell, 1996] Chiang, W. C. and Russell, R. A. (1996). Simulated annealing metaheuristics for the vehicle routing problem with time windows. *Annals of Operations Research*.

[Christofides, 1985] Christofides, N. (1985). Vehicle Routing. In Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G., and Shmoys, D. B., editors, *The Traveling Salesman Problem. A Guided Tour of Combinatorial Optimization*, chapter Vehicle Ro, pages 431–448. Wiley, Chichester.

[Christofides et al., 1981] Christofides, N., Mingozzi, A., and Toth, P. (1981). Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. *Mathematical Programming*, 20(1):255–282.

[Clarke and Wright, 1964] Clarke, G. and Wright, J. W. (1964). Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operations Research*, 12(4):568–581.

[Croes, 1958] Croes, G. A. (1958). A Method for Solving Traveling-Salesman Problems. *Operations Research*, 6(6):791–812.

[Dantzig and Ramser, 1959] Dantzig, G. B. and Ramser, J. H. (1959). The Truck Dispatching Problem. *Management Science*, 6(1):80–91.

[De Santiago et al., 2013] De Santiago, A., Melián-Batista, B., Álvarez, A., and AngelBello, F. (2013). Effect of the Initial Solutions to Balance Routes in Vehicle

Routing Problem with Time Windows. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pages 46–52.

[Dror and Trudeau, 1990] Dror, M. and Trudeau, P. (1990). Split delivery routing. *Naval Research Logistics (NRL)*.

[El-Sherbeny, 2010] El-Sherbeny, N. A. (2010). Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods. *Journal of King Saud University - Science*, 22(3):123–131.

[Gaskell, 1967] Gaskell, T. J. (1967). Bases for Vehicle Fleet Scheduling. *OR*, 18(3):281.

[Gendreau et al., 1994] Gendreau, M., Hertz, A., and Laporte, G. (1994). A Tabu Search Heuristic for the Vehicle Routing Problem. *Management Science*, 40(10):1276–1290.

[Ghiani et al., 2013] Ghiani, G., Laporte, G., and Musmanno, R. (2013). *Introduction to Logistics Systems Management*. John Wiley & Sons, Ltd, Chichester, UK.

[Glover, 1977] Glover, F. (1977). HEURISTICS FOR INTEGER PROGRAMMING USING SURROGATE CONSTRAINTS. *Decision Sciences*, 8(1):156–166.

[Glover, 1998] Glover, F. (1998). A template for scatter search and path relinking. pages 1–51.

[Kumar and Panneerselvam, 2012] Kumar, S. N. and Panneerselvam, R. (2012). A Survey on the Vehicle Routing Problem and Its Variants. *Intelligent Information Management*, 04(03):66–74.

[Landeghem, 1988] Landeghem, H. (1988). A bi-criteria heuristic for the vehicle routing problem with time windows. *European Journal of Operational Research*, 36(2):217–226.

[Laporte, 1992] Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(3):345–358.

[Lee and Ueng, 1999] Lee, T. and Ueng, J. (1999). A study of vehicle routing problems with loadbalancing. *International Journal of Physical Distribution & Logistics Management*, 29(10):646–657.

[Lenstra and Kan, 1975] Lenstra, J. K. and Kan, A. H. G. R. (1975). Some Simple Applications of the Travelling Salesman Problem. *Operational Research Quarterly (1970-1977)*, 26(4):717.

[Lozano et al., 2018] Lozano, J., González-Gurrola, L.-C., Rodríguez-Tello, E., and Lacomme, P. (2018). A statistical comparison of objective functiorn for the vehicle routing problem with route balancing. In *2016 Fifteenth Mexican International Conference on Artificial Intelligence (MICAI)*. IEEE.

[Martí et al., 2006] Martí, R., Laguna, M., and Glover, F. (2006). Principles of scatter search. *European Journal of Operational Research*, 169(2):359–372.

[Matl, P., Hartl, R. F., & Vidal, 2017] Matl, P., Hartl, R. F., & Vidal, T. (2017). Workload equity in vehicle routing problems: A survey and analysis. *Transportation Science*, 52(2):239–260.

[Paessens, 1988] Paessens, H. (1988). The savings algorithm for the vehicle routing problem. *European Journal of Operational Research*, 34(3):336–344.

[Potvin and Rousseau, 1995] Potvin, J. Y. and Rousseau, J.-M. (1995). An exchange heuristic for routing problems with time windows. *Journal of the Operational Research Society*, 46(12):1433–1466.

[Sivaramkumar et al., 2018] Sivaramkumar, V., Thansekhar, M. R., Saravanan, R., and Miruna Joe Amali, S. (2018). Demonstrating the importance of using total time balance instead of route balance on a multi-objective vehicle routing problem with time windows. *The International Journal of Advanced Manufacturing Technology*, 98(5-8):1287–1306.

[Solomon, 2008] Solomon, M. M. (2008). Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. *Operations Research*.

[Swersey and Ballard, 2008] Swersey, A. J. and Ballard, W. (2008). Scheduling School Buses. *Management Science*.

[Taillard et al., 1997] Taillard, É., Badeau, P., Gendreau, M., Guertin, F., and Potvin, J.-Y. (1997). A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows. *Transportation Science*, 31(2):170–186.

[Thangiah, 1995] Thangiah, S. R. (1995). Vehicle Routing with Time Windows using Genetic Algorithms. In Chambers, L., editor, *Application Handbook of Genetic Algorithms: New Frontiers*, volume 2, pages 253–277. CRC Press.

[Toth and Vigo, 2002a] Toth, P. and Vigo, D. (2002a). Models, relaxations and exact approaches for the capacitated vehicle routing problem. *Discrete Applied Mathematics*, 123(1-3):487–512.

[Toth and Vigo, 2002b] Toth, P. and Vigo, D. (2002b). *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics.

[Vega-Mejía et al., 2019] Vega-Mejía, C. A., Montoya-Torres, J. R., and Islam, S. M. N. (2019). Consideration of triple bottom line objectives for sustainability in the optimization of vehicle routing and loading operations: a systematic literature review. *Annals of Operations Research*, 273(1-2):311–375.

[Wren and Holliday, 1972] Wren, A. and Holliday, A. (1972). Computer Scheduling of Vehicles from One or More Depots to a Number of Delivery Points. *Journal of the Operational Research Society*, 23(3):333–344.

[Yellow, 1970] Yellow, P. C. (1970). A Computational Modification to the Savings Method of Vehicle Scheduling. *Operational Research Quarterly (1970-1977)*, 21(2):281.

[Zhou et al., 2013] Zhou, W., Song, T., He, F., and Liu, X. (2013). Multiobjective Vehicle Routing Problem with Route Balance Based on Genetic Algorithm. *Discrete Dynamics in Nature and Society*, 2013:1–9.

Appendix A

DATA SETS

Table A.1: Customer Information of Data Set 1-2

Customer	Service Time	Demand	Latitude	Longitude	Earliest Time	Latest Time
1	18	6	40.212381	28.98842	930	1020
2	24	8	40.21083	28.995846	810	1020
3	33	11	39.4363	29.994994	690	780
4	24	8	39.421054	29.986327	750	840
5	18	6	40.084156	29.510051	930	1020
6	18	6	40.20718	28.995781	750	960
7	21	7	40.220002	28.978378	720	810
8	18	6	40.20718	28.995781	750	960
9	18	6	40.2105	28.9947	810	1020
10	24	8	40.084354	29.510399	930	1020
11	15	5	40.20718	28.995781	750	960
12	21	7	40.210688	28.994675	810	1020
13	15	5	40.20718	28.995781	750	960
14	39	13	40.840675	31.152629	90	180
15	21	7	40.721384	29.819633	210	300
16	27	9	40.764232	29.926204	90	180
17	18	6	40.760741	29.934072	390	480
18	27	9	40.788954	30.377352	270	360
19	24	8	40.792358	30.38493	90	180
20	18	6	40.788954	30.377352	270	360
21	18	6	40.764586	29.940533	270	360
22	24	8	40.763458	29.928558	120	210
23	21	7	40.840573	31.155957	210	300
24	15	5	40.757196	29.945358	450	540
25	24	8	41.455675	31.764206	90	240
26	24	8	40.777766	30.364746	150	240
27	15	5	41.454956	31.764184	150	240
28	18	6	40.774982	30.38091	90	180
29	15	5	40.77774	30.364912	120	210
30	15	5	41.455871	31.764206	150	240
31	15	5	40.757196	29.945358	450	540
32	15	5	40.760159	29.934092	390	480

Table A.2: Customer Information of Data Set 3-18

Customer	Service Time	Demand	Latitude	Longitude	Earliest Time	Latest Time
1	20	0.05	41.15984	29.02523	240	480
2	20	0.303	41.15984	29.02523	240	480
3	20	0.364	41.06708	29.02737	240	480
4	20	0.39	41.12721	28.61469	240	480
5	20	0.305	41.12721	28.61469	240	480
6	20	1.219	41.0062	28.82804	240	480
7	20	0.053	41.0062	28.82804	240	480
8	20	0.379	41.0062	28.82804	240	480
9	20	0.123	41.0062	28.82804	240	480
10	20	0.39	41.0062	28.82804	240	480
11	20	0.307	40.9228	29.30512	480	720
12	20	0.375	40.9228	29.30512	480	720
13	20	0.303	41.14149	28.46621	240	720
14	20	0.053	41.14149	28.46621	240	720
15	20	0.373	40.95065	29.10072	240	480
16	20	0.389	41.05446	28.87238	240	480
17	20	1.554	40.98033	29.04264	240	480
18	20	0.051	40.90772	29.17363	480	720
19	20	0.294	40.97512	28.73239	480	720
20	20	0.389	40.96942	28.65593	240	480
21	20	1.554	40.96942	28.65593	240	480
22	20	0.367	40.96942	28.65593	240	480
23	20	1.203	41.01676	28.60715	480	720
24	20	0.379	41.14173	29.14546	480	720
25	20	0.379	41.05991	28.91528	240	480
26	20	0.379	41.03891	29.11044	480	720
27	20	1.376	41.03891	29.11044	480	720
28	20	0.376	41.03891	29.11044	480	720
29	20	0.144	41.04892	28.97966	240	480
30	20	1.563	40.92875	29.21162	240	480
31	20	0.381	40.89387	29.22834	240	480
32	20	1.554	41.11703	28.76134	240	480
33	20	0.387	40.97703	29.08672	240	480
34	20	0.369	40.97334	29.06671	240	480
35	20	0.069	40.88687	29.19829	240	480

Customer Information of Data Set 3-18 (Continued)

Customer	Service Time	Demand	Latitude	Longitude	Earliest Time	Latest Time
36	20	1.204	40.93129	29.21981	480	720
37	20	0.373	40.80323	29.42661	240	720
38	20	0.367	41.08635	28.98083	240	720
39	20	0.367	41.08635	28.98083	240	720
40	20	0.367	41.08635	28.98083	240	720
41	20	0.381	41.03485	29.08152	240	480
42	20	0.305	41.03485	29.08152	240	480
43	20	0.053	41.03485	29.08152	240	480
44	20	0.145	41.03485	29.08152	240	480
45	20	0.389	41.03485	29.08152	240	480
46	20	0.398	40.95819	29.07821	240	480
47	20	1.554	40.95819	29.07821	240	480
48	20	1.563	41.04665	28.04126	240	720
49	20	0.367	41.1209	29.13589	480	720
50	20	0.052	41.02286	29.02023	240	480
51	20	0.283	41.02462	29.01754	240	720
52	20	1.376	41.01452	29.05451	480	720
53	20	0.395	40.93346	29.16404	480	720
54	20	0.376	40.93346	29.16404	480	720
55	20	0.127	41.01538	28.86883	240	480
56	20	1.117	41.0505	29.0232	240	480
57	20	0.374	41.16633	29.0389	480	720
58	20	0.379	40.97955	28.6357	240	480
59	20	1.205	40.97955	28.6357	240	480
60	20	0.39	40.97955	28.6357	240	480
61	20	0.375	41.05952	28.6752	240	480
62	20	0.381	41.02061	28.63946	240	480
63	20	1.205	40.9859	28.78176	240	720
64	20	1.124	41.05673	28.68191	240	480
65	20	0.358	41.00407	29.0618	240	480
66	20	0.362	40.89613	29.1714	240	480
67	20	0.383	41.02925	28.8381	240	480
68	20	1.345	41.02925	28.8381	240	480
69	20	0.385	41.02925	28.8381	240	480
70	20	1.369	41.07951	29.04806	240	480

Customer Information of Data Set 3-18 (Continued)

Customer	Service Time	Demand	Latitude	Longitude	Earliest Time	Latest Time
71	20	0.069	40.99264	28.54004	240	480
72	20	0.05	41.06113	28.92934	240	480
73	20	0.298	41.06113	28.92934	240	480
74	20	0.057	41.06113	28.92934	240	480
75	20	0.091	41.06664	28.69232	240	480
76	20	0.381	40.98546	29.12224	480	720
77	20	0.376	40.92639	29.14444	720	840
78	20	0.367	41.01217	28.60696	240	480
79	20	0.05	40.99197	28.90634	240	480
80	20	0.298	40.99197	28.90634	240	480
81	20	1.376	40.96118	29.22202	240	480
82	20	1.346	41.01718	28.94292	240	480
83	20	0.144	41.01718	28.94292	240	480
84	20	0.05	41.01718	28.94292	240	480
85	20	0.389	41.01718	28.94292	240	480
86	20	0.296	41.01718	28.94292	240	480
87	20	0.427	40.97091	29.08179	240	480
88	20	0.427	40.96962	29.08927	240	480
89	20	1.376	40.96962	29.08927	240	480
90	20	1.331	41.08578	28.97756	480	720
91	20	0.296	40.81959	29.44223	240	720
92	20	0.398	41.07677	29.02885	240	480
93	20	0.378	41.21608	28.7644	480	720
94	20	0.381	41.21608	28.7644	480	720
95	20	1.563	41.21608	28.7644	480	720
96	20	0.376	41.05944	28.90152	480	720
97	20	1.34	41.05944	28.90152	480	720
98	20	0.051	41.05944	28.90152	480	720
99	20	0.383	41.05944	28.90152	480	720
100	20	0.144	41.05944	28.90152	480	720
101	20	0.298	41.05944	28.90152	480	720
102	20	0.376	40.78239	29.37286	240	720
103	20	1.393	41.04285	28.83566	480	720
104	20	0.39	41.04285	28.83566	480	720
105	20	0.395	40.93503	29.15275	240	480

Customer Information of Data Set 3-18 (Continued)

Customer	Service Time	Demand	Latitude	Longitude	Earliest Time	Latest Time
106	20	0.387	41.00481	28.83545	480	720
107	20	0.127	41.00404	28.62118	240	480
108	20	0.379	40.91311	29.20699	240	480
109	20	0.057	41.20974	29.02162	240	480
110	20	0.381	41.02038	29.02499	240	480
111	20	0.381	41.02833	29.09317	240	480
112	20	0.39	41.02833	29.09317	240	480
113	20	0.374	41.08535	28.62186	480	720
114	20	1.117	41.08535	28.62186	480	720
115	20	0.846	41.07179	28.6769	480	720
116	20	0.218	41.07179	28.6769	480	720
117	20	0.116	41.07179	28.6769	480	720
118	20	0.376	40.9575	29.28774	480	720
119	20	1.208	40.9575	29.28774	480	720
120	20	0.369	41.0325	28.98332	240	480
121	20	0.051	41.02372	28.8587	480	720
122	20	0.144	41.02372	28.8587	480	720
123	20	0.293	41.02372	28.8587	480	720
124	20	1.554	40.99547	28.7716	240	480
125	20	0.376	41.13978	28.46119	240	720
126	20	0.376	41.02159	28.65318	240	480
127	20	1.337	41.06778	29.04417	240	480
128	20	0.378	41.0072	29.07342	240	480
129	20	0.069	41.06497	28.99566	240	480
130	20	0.204	41.06497	28.99566	240	480
131	20	0.389	41.06497	28.99566	240	480
132	20	1.529	41.07482	28.99435	480	720
133	20	1.313	41.00766	28.88731	480	720
134	20	0.372	41.00388	28.86822	480	720
135	20	1.365	41.00388	28.86822	480	720
136	20	1.376	41.07785	28.88225	240	480
137	20	0.363	40.99851	28.64946	480	720
138	20	0.145	40.99851	28.64946	480	720
139	20	0.051	40.99851	28.64946	480	720
140	20	0.296	40.99851	28.64946	480	720

Customer Information of Data Set 3-18 (Continued)

Customer	Service Time	Demand	Latitude	Longitude	Earliest Time	Latest Time
141	20	0.951	40.98353	28.85366	240	480
142	20	0.387	40.91379	29.16092	240	480
143	20	0.398	40.97109	29.08215	720	840
144	20	0.367	41.02416	28.94715	240	480
145	20	0.374	40.8956	29.19631	240	720
146	20	0.389	40.98121	29.08277	240	480
147	20	1.205	40.98121	29.08277	240	480
148	20	0.381	40.91296	29.25	240	480
149	20	0.387	40.99775	28.78276	240	480
150	20	1.339	41.00857	29.21007	480	720
151	20	0.155	41.06974	29.00942	240	480
152	20	0.367	41.05762	28.87187	240	480
153	20	0.379	41.01241	29.13272	240	480
154	20	0.381	41.10383	28.76353	480	720
155	20	0.373	40.81509	29.38662	240	720
156	20	1.376	40.97803	28.63546	240	480
157	20	0.39	40.97803	28.63546	240	480
158	20	0.376	41.06411	28.70994	240	480
159	20	0.39	41.01614	28.84621	720	840
160	20	0.389	40.9259	29.29835	240	480
161	20	0.367	40.9259	29.29835	240	480
162	20	0.396	40.87505	29.28764	240	480
163	20	0.381	41.07117	28.93827	240	480
164	20	0.379	40.86811	29.29297	240	480
165	20	0.387	41.03715	28.85338	480	720
166	20	0.381	40.85924	29.27998	240	480
167	20	0.296	41.08019	28.94304	240	480
168	20	0.052	41.08019	28.94304	240	480
169	20	0.366	40.92323	29.29257	480	720
170	20	0.376	40.99234	28.90254	240	480
171	20	1.203	40.99594	28.78472	480	720
172	20	0.39	40.90988	29.22324	240	480
173	20	0.38	41.07726	28.95347	480	720
174	20	1.196	41.0806	28.66788	480	720
175	20	0.373	40.93972	29.31434	240	480

Customer Information of Data Set 3-18 (Continued)

Customer	**Service Time**	**Demand**	**Latitude**	**Longitude**	**Earliest Time**	**Latest Time**
176	20	0.145	41.05532	28.78012	240	480
177	20	1.349	41.01558	28.92997	720	840
178	20	0.375	40.98758	28.67697	720	840
179	20	0.373	41.00952	28.66623	240	480
180	20	0.375	40.99162	28.79012	720	840
181	20	0.375	41.02217	29.15362	240	480
182	20	0.376	40.96242	29.07101	240	480
183	20	1.365	40.98425	29.13543	240	480
184	20	0.385	40.87824	29.28357	240	480
185	20	1.224	41.00736	29.20304	480	720
186	20	1.114	40.82548	29.37645	240	720
187	20	1.22	41.2686	28.69409	240	480
188	20	0.296	41.0372	28.82706	240	720
189	20	0.376	41.00626	29.09109	240	480
190	20	0.387	40.98311	29.22737	240	480
191	20	0.05	40.98311	29.22737	240	480
192	20	0.382	40.98311	29.22737	240	480
193	20	0.301	40.98311	29.22737	240	480
194	20	0.057	40.98311	29.22737	240	480
195	20	1.213	40.98311	29.22737	240	480
196	20	0.376	41.0372	28.82706	240	720
197	20	1.19	41.06994	28.89981	240	480
198	20	0.378	41.07491	28.97723	240	720
199	20	0.116	40.99529	29.13829	240	480
200	20	0.373	40.98904	29.22898	480	720
201	20	1.34	40.98904	29.22898	480	720
202	20	0.383	40.98904	29.22898	480	720
203	20	0.374	40.91461	29.30157	240	480
204	20	0.39	41.01567	29.12569	240	480
205	20	0.398	40.99809	28.76423	240	480
206	20	0.389	40.99809	28.76423	240	480
207	20	1.376	40.99809	28.76423	240	480
208	20	0.376	40.99809	28.76423	240	480
209	20	0.381	41.08565	29.00039	240	480
210	20	0.357	40.97583	28.79552	240	480

Customer Information of Data Set 3-18 (Continued)

Customer	**Service Time**	**Demand**	**Latitude**	**Longitude**	**Earliest Time**	**Latest Time**
211	20	1.376	41.01753	29.1045	240	480
212	20	0.382	40.98163	28.88252	240	480
213	20	0.305	40.84001	29.308	240	480
214	20	0.053	40.84001	29.308	240	480
215	20	1.224	41.10184	28.88045	240	480
216	20	0.144	40.99121	28.62433	240	480
217	20	0.05	40.99121	28.62433	240	480
218	20	0.366	41.12406	29.05071	480	720
219	20	0.381	40.98119	29.08788	240	720
220	20	0.379	41.03432	28.89263	240	480
221	20	1.196	41.03432	28.89263	240	480
222	20	0.383	41.06895	28.90735	240	480
223	20	0.144	40.8923	29.20052	240	480
224	20	0.05	40.8923	29.20052	240	480
225	20	0.374	40.8923	29.20052	240	480
226	20	0.372	40.8923	29.20052	240	480
227	20	1.22	41.07757	28.98204	240	480
228	20	1.331	41.06249	28.89591	240	480
229	20	0.387	41.01154	29.14301	240	480
230	20	0.353	41.11064	29.04012	480	720
231	20	1.34	41.08205	28.87881	240	480
232	20	0.05	41.08205	28.87881	240	480
233	20	0.39	41.08205	28.87881	240	480
234	20	0.303	41.08205	28.87881	240	480
235	20	1.117	40.87734	29.23543	240	480
236	20	0.379	40.97482	29.23495	240	480
237	20	1.376	40.97482	29.23495	240	480
238	20	1.361	40.87603	29.22646	240	480
239	20	0.053	40.981533	29.234034	480	720
240	20	0.381	40.915403	29.204291	480	720
241	20	0.144	40.96555	29.07598	240	480
242	20	1.554	40.96555	29.07598	240	480
243	20	0.398	40.90986	29.16209	240	720
244	20	0.381	40.99494	28.52572	240	480
245	20	0.381	41.14014	28.83537	240	480

Customer Information of Data Set 3-18 (Continued)

Customer	Service Time	Demand	Latitude	Longitude	Earliest Time	Latest Time
246	20	0.439	41.06719	28.89599	240	480
247	20	0.051	40.98779	29.07821	240	480
248	20	0.301	40.98779	29.07821	240	480
249	20	1.19	40.92626	29.27344	240	480
250	20	1.209	41.01199	28.64515	240	480
251	20	0.39	40.99043	29.16361	720	840
252	20	0.373	41.24698	29.03671	240	480
253	20	0.292	41.01995	29.02219	240	480
254	20	0.155	41.02546	28.66132	240	480
255	20	0.381	41.00909	28.54671	240	480
256	20	0.381	41.10979	28.67895	240	480
257	20	0.376	41.01329	28.8223	240	480
258	20	0.053	41.0111	28.94605	240	480
259	20	0.374	41.03043	28.69631	240	480
260	20	0.385	40.82566	29.30323	720	840
261	20	0.395	41.04892	28.97966	240	480
262	20	0.296	41.03237	28.89361	720	840
263	20	0.052	41.03237	28.89361	720	840
264	20	0.371	40.96194	28.81894	240	480
265	20	0.367	40.96194	28.81894	240	480
266	20	1.205	40.98587	29.10963	240	480
267	20	0.387	40.93463	29.12922	240	480
268	20	0.057	41.0372	28.82706	240	720
269	20	1.222	40.91461	29.30157	240	480
270	20	0.069	41.05788	28.85504	240	480
271	20	0.398	41.10225	29.04875	480	720
272	20	1.092	41.01591	28.9384	480	720
273	20	1.361	41.16803	29.03307	240	480
274	20	1.22	41.04213	28.86286	240	480
275	20	0.367	41.017	28.85912	480	720
276	20	0.398	41.04403	29.0099	480	720
277	20	0.376	41.04403	29.0099	480	720
278	20	0.069	41.0852	29.04654	240	480
279	20	0.069	41.0852	29.04654	240	480
280	20	0.069	41.0852	29.04654	240	480

Customer Information of Data Set 3-18 (Continued)

Customer	Service Time	Demand	Latitude	Longitude	Earliest Time	Latest Time
281	20	0.396	41.06104	28.60068	240	480
282	20	0.05	40.99708	29.26361	480	720
283	20	1.554	41.02877	29.05635	240	480
284	20	0.378	41.05912	28.96158	240	480
285	20	1.376	41.0493	29.06294	240	480
286	20	1.203	41.04121	28.8342	240	480
287	20	0.358	41.03655	28.7879	240	480
288	20	1.22	40.96857	29.2511	480	720
289	20	1.208	41.01243	28.83383	240	480
290	20	1.117	40.8512	29.301	240	480
291	20	0.367	41.006242	28.659081	240	720
292	20	0.389	40.97974	29.06583	480	720
293	20	0.389	41.01258	28.6098	240	480
294	20	0.383	41.01258	28.6098	240	480
295	20	0.387	40.96144	28.82835	240	480
296	20	1.549	41.0968	29.08714	240	480
297	20	0.358	40.99988	29.03245	240	480
298	20	0.374	41.03502	29.05755	240	480
299	20	1.114	41.03822	28.88011	720	840
300	20	0.375	41.02027	28.87244	240	480
301	20	0.145	41.02027	28.87244	240	480
302	20	0.051	41.02027	28.87244	240	480
303	20	1.205	41.02027	28.87244	240	480
304	20	0.289	41.02027	28.87244	240	480
305	20	1.22	41.05052	28.84211	240	480
306	20	0.374	40.99313	28.78583	240	480
307	20	0.384	41.01233	28.84105	240	480
308	20	0.38	41.03436	28.68281	240	480
309	20	1.19	40.9847	29.02621	240	480
310	20	0.057	41.06916	28.89989	240	480
311	20	0.378	41.00414	28.77385	240	480
312	20	1.554	40.98771	28.86821	240	480
313	20	0.303	41.00251	28.90565	480	720
314	20	0.387	41.03969	28.77283	240	480
315	20	1.563	41.03969	28.77283	240	480

Customer Information of Data Set 3-18 (Continued)

Customer	Service Time	Demand	Latitude	Longitude	Earliest Time	Latest Time
316	20	1.092	41.06377	29.06867	240	480
317	20	0.398	40.9755	28.8748	480	720
318	20	0.367	40.9755	28.8748	480	720
319	20	0.39	40.98279	29.24319	240	480
320	20	0.301	41.02219	29.01477	240	480
321	20	0.051	41.02219	29.01477	240	480
322	20	0.144	41.02219	29.01477	240	480
323	20	0.374	41.02219	29.01477	240	480
324	20	1.22	41.10955	28.87331	720	840
325	20	0.358	41.17651	28.89066	240	480
326	20	0.381	41.11883	29.09819	240	480
327	20	0.381	40.97947	29.04922	240	480
328	20	0.387	40.97947	29.04922	240	480
329	20	0.438	41.17064	29.05963	720	840
330	20	1.376	41.179145	28.887136	240	720
331	20	0.387	41.02651	28.67353	480	720
332	20	0.373	41.04922	28.8213	240	480
333	20	0.381	41.17707	28.88981	240	480
334	20	1.563	41.00909	29.10257	480	720
335	20	0.381	40.994507	29.08878	240	480
336	20	0.381	41.07831	28.96317	240	480
337	20	0.381	41.03003	29.10093	240	480
338	20	1.331	41.03906	28.70092	240	480
339	20	0.387	41.04812	28.99692	480	720
340	20	0.387	41.07345	29.044	240	480

Appendix B

COMPARISON TABLES

Table B.1: Comparison Table of Exact Models (in terms of total time)

	Sivaramkumar et al. Exact Model					**VRPTWRB Exact Model**				
Data set	**Total time (min)**	**Total number of vehicles**	**CPU time (sec)**	**% Gap**	**Average Route End Time(min)**	**Total time (min)**	**Total number of vehicles**	**CPU time (sec)**	**% Gap**	**Average Route End Time(min)**
1(C13V27)	1021	6	14	0%	856 ±82.42	1021	6	12	0%	**844 ±69.85**
2(C19V27)	1473	10	14	0%	893 ±72.69	1473	10	51	0%	**852 ±62.49**
3(C11V25)	501.9	4	8700	0%	548.7 ±157.5	709.433	6	4	0%	**470 ±170.3**
4(C5V25)	282.45	3	1	0%	468.6 ±220.4	282.45	3	1	0%	**308.3 ±11.78**
5(C12V25)	453.417	4	16	0%	672.2 ±107.7	618.333	5	304	0%	**555.4 ±20.21**
6(C12V25)	587.867	4	14	0%	623 ±90.77	830.333	6	26	0%	**634.1 ±16.9**
7(C14V25)	234.967	6	26	0%	407.6 ±115.2	253.533	7	36	0%	**387.2 ±108.7**
8(C22V25)	931.983	8	86400*	33.39%	-	1117.23	10	86400*	23.58%	-
9(C17V25)	577.183	6	86400*	10.85%	-	825.417	9	86400*	4.34%	-
10(C11V25)	692.9	5	16	0%	550 ±143.5	831.067	6	7	0%	**437.6 ±110.2**
11(C22V25)	1444.1	9	86400*	33.56%	-	668.5	8	86400*	5.93%	-
12(C14V25)	694.6	11	86400*	58.70%	-	784.083	7	86400*	4.91%	-
13(C14V25)	747.85	5	86400*	48.30%	-	892.35	6	4080	0%	**469.8 ±99.23**
14(C16V25)	817.817	6	86400*	49%	-	840.4	7	4320	0%	**540 ±6.45**

* Time limit reached.

Table B.2: Comparison Table of Exact Models (in terms of $L_{max} - L_{min}$)

	Sivaramkumar et al. Exact Model					**VRPTWRB Exact Model**				
Data set	**Obj. (min)**	**Total number of vehicles**	**CPU time (sec)**	**% Gap**	**Average Route End Time(min)**	**Obj. (min)**	**Total number of vehicles**	**CPU time (sec)**	**% Gap**	**Average Route End Time(min)**
1(C13V27	339	6	600	0%	963.4 ±94.74	48	8	**16**	0%	**934.5 ±22.12**
2(C19V27	258	17	86400*	3.17%	816.5 ±186.2	396	**13**	**5**	0%	**888.12 ±119.9**
3(C11V25	115	11	18	0.00%	474.5 ±179.9	243	**7**	**3**	0%	**642.7 ±118.8**
4(C20V25	22	20	3645	0%	402.8 ±118.7	6	**14**	**31**	0%	**541.0 ±3.213**
5(C5V25	138	6	2	0%	300 ±16	0	**3**	**1**	0%	**765 ±18.45**
6(C12V25	125	12	29	0%	400 ±118.3	16.717	**8**	61	0%	**545.6 ±8.260**
7(C12V25	1417	6	5340	0%	512.8 ±155.6	238	6	**2**	0%	**588.3 ±87.63**
8(C14V25	14	10	86400*	54.59%	-	19.083	**8**	**27**	0%	**542.8 ±8.623**
9(C22V25	13	20	456	0%	540 ±195.9	275.3	**10**	86400*	11.01%	-
10(C17V25	9.867	10	10140	0%	519.5 ±162.9	237	10	**6**	0%	**576.8 ±71.53**
11(C11V25	18	10	3	0%	485 ±194.6	0	**8**	**2**	0%	**540.7 ±2.384**
12(C14V25	103.25	13	141	0%	427 ±113.4	19.683	**8**	**17**	0%	**529 ±15.04**
13(C14V25	150	13	7226	0%	492 ±179.5	24.483	**8**	**23**	0%	**547.2 ±12.07**
14(C16V25	123	15	117	0%	389.6 ±106.9	0	**12**	**49**	0%	**542.8 ±4.740**
15(C22V25	36	15	8	0%	544.4 ±152.0	251	**13**	458	0%	**562 ±63.88**
16(C22V25	7.833	20	35	0%	521.6 ±137.3	242	**12**	64	0%	**609.3 ±99.46**
17(C28V25	68	27	86400*	19.60%	-	237	**16**	**226**	0%	**540.1 ±25.8**
18(C34V25	45	29	86400*	70.66%	-	243	**22**	**4680**	0%	**531.8 ±22.2**

* Time limit reached.

Table B.3: Comparison Table of Exact Models (in terms of total time)

	Belfiore and Yoshizaki Exact Model					**SDVRPTWRB Exact Model**				
Data set	**Total time (min)**	**Total number of vehicles**	**CPU time (sec)**	**% Gap**	**Average Route End Time(min)**	**Total time (min)**	**Total number of vehicles**	**CPU time (sec)**	**% Gap**	**Average Route End Time(min)**
1(C13V27)	2474	7	3	0%	819.4 ±95.57	2474	7	6	0%	**819.8 ±88.4**
2(C19V27)	2278	7	7	0%	926.6 ±140.3	2552	9	47	0%	**816.4 ±68.9**
3(C11V25)	2089.17	6	1	0%	530.1 ±195.4	2089.89	6	2	0%	**510.1 ±159.4**
4(C5V25)	456	5	1	0%	300.0 ±0	456	5	1	0%	300.0 ±0
5(C12V25)	109.25	1	86400*	18.5%	-	269.42	5	59	0%	594 ±137.7
6(C12V25)	800.45	5	1	0%	560.0 ±223.6	811.6	6	1	0%	**512.5 ±209.2**
7(C14V25)	235.75	7	1	0%	505.7 ±153.3	261.47	8	3	0%	**399.2 ±109.4**
9(C17V25)	877.05	9	2	0%	538.2 ±201.8	880.64	10	8	0%	**547.0 ±107.8**
10(C11V25)	1313.7	6	1	0%	503.8 ±166.0	1320.19	7	4	0%	**444.2 ±117.6**
11(C22V25)	1569.69	10	5	0%	602.3 ±219.8	1569.69	10	660	0%	**455.8 ±159.9**
12(C14V25)	1727.15	5	1	0%	645.6 ±111.1	1766.51	8	61	0%	**461.7 ±107.2**
13(C14V25)	1389.82	5	4	0%	468.0 ±144	1403.64	7	7	0%	**481.4 ±115.1**
14(C16V25)	1092.73	4	3	0%	540.0 ±240	1133.73	8	27	0%	**432.2 ±126.9**
18(C20V25)	377.39	10	2	0%	404.8 ±124.3	391.467	10	9	0%	**430.8 ±121.5**

* Time limit reached.