

MASTER

Vehicle routing problem with flexible time windows

Arslantay, E.

Award date:
2011

[Link to publication](#)

Disclaimer

This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

Eindhoven, August 2011

Vehicle Routing Problem with Flexible Time Windows

by

Ezgi Arslantay

BSc Industrial Engineering – Bilkent University, 2009

Student identity number 0728342

in partial fulfillment of the requirements for the degree of

Master of Science

in Operations Management and Logistics

Supervisors:

prof. dr. Tom Van Woensel, TU/e, OPAC

dr. Ola Jabali, École Polytechnique de Montréal

TUE . School of Industrial Engineering

Series Master Thesis Operations Management and Logistics

Subject headings: vehicle routing problems with flexible time windows, tabu search, metaheuristics, allowance for violation of customer time windows, time oriented nearest neighbor heuristic, vehicle scheduling, linear programming

Abstract

This master thesis describes a research project conducted in the field of vehicle routing problem with flexible time window constraints (VRPFTW), in which vehicles are allowed to start servicing customers before and after the earliest and latest time window bounds, respectively. The time windows are often relaxed to allow for early or late arrivals at customer locations. That relaxation comes at the penalty costs as the time window violations has an effect on the customers' satisfaction. However, in applications where increasing customer satisfaction level is much more important, the penalty for earliness or tardiness needs to be minimized. The objective of this problem is to assign vehicles to feasible routes and make schedules that minimize the total costs, including travel cost, expected penalty cost for earliness or tardiness and cost of used vehicles. In this thesis, we describe an algorithm to solve the problem, which develops a Tabu Search heuristic incorporated with a linear programming. Three different approximation approaches are provided as well as the exact evaluation. The algorithm is tested on a number of benchmark instances and provided good quality solutions.

Preface

This document presents the results of my master thesis project to complete the MSc. Operations Management and Logistics program at Eindhoven University of Technology. This project was carried out for five months from February 2011 to August 2011. In the following paragraphs, I would like to thank people who helped me throughout this project.

First of all, I would like to thank my first supervisor at TU/e, Tom van Woensel, for his contribution, support and guidance. His extensive knowledge and experience enabled me to improve many aspects of this project. I appreciate his positive attitude, endless patience and tolerance on me. Additionally, I would like to thank my second supervisor, Ola Jabali, who has been through all the stages of this thesis, for her constant encouragement and guidance. Without her consistent instruction and invaluable help, this thesis would not have reached its present form.

Besides, I am grateful to all my friends. They motivated me during the difficult moments, helped me through the tough time and shared my unforgettable moments during my study in Eindhoven.

Last but not least, I would like to thank the people that are most valuable, my family. I must thank my mother and my father for encouraging me to come to the Netherlands and spend the international semester in South Korea, for supporting me continuously and most importantly, for being there to help me whenever I need.

Ezgi Arslantay,

Eindhoven, August 2011

Executive Summary

The Vehicle Routing Problem (VRP) has drawn a lot of attention because of its key role in improving the efficiency of distribution and decreasing the transportation cost. The VRP aims at designing a set of vehicle routes through several customer locations with minimum costs, under the conditions that each route starts and ends at the depot and each customer must be visited only once by one vehicle.

The problem we consider in this study is called Vehicle Routing Problem with Flexible Time Windows (VRPFTW) in which vehicles are allowed to start servicing customers before and after the earliest and latest time window bounds, respectively. The time windows are often relaxed to allow for early or late arrivals at customer locations. That relaxation comes at the penalty costs as the time window violations has an effect on the customers' satisfaction. Violation is defined as the early or late arrival to the particular customer location at a cost of a penalty proportional to the extension in the time window and must be penalized to reflect the negative effects of customer satisfaction.

The objective of this problem is to assign vehicles to feasible routes and make schedules that minimize the total costs, including travel cost, expected penalty cost for earliness or tardiness and cost of used vehicles. In this thesis, we describe an algorithm to solve the problem, which develops a Tabu Search heuristic incorporated with a linear programming. Three different approximation approaches are provided as well as the exact evaluation. The algorithm is tested on a number of benchmark instances and provided good quality solutions.

Several exact and approximate algorithms have been proposed to get the solutions of variants of VRP, but NP-hardness of those kinds of problem settings requires heuristic solution strategies for most real-life instances. The algorithm we propose, adopts the concepts of Tabu Search, but incorporates VRPFTW specific features. We construct a linear programming (LP) model to make a robust schedule. The objective function of the LP is used to exactly evaluate solution. However, computing the objective function for all candidate solutions is an expensive approach; thus an approximation function has to be used to evaluate possible neighbors of a given solution and choose the best one as a new current solution.

The algorithm for VRPFTW follows the listed steps:

1. The route planning: The initial solution is found through the time-oriented Nearest Neighbor Heuristic. After Or-opt and 2-opt* exchange operators generate the neighborhood, all the

candidates are evaluated using the one of three approximation function, from which the one with the best value is selected as the current solution. Afterwards, the exact function is computed to evaluate the current solution. The tabu list collects the best moves from each of the iterations and the aspiration criteria forces the exact value of a forbidden solution to be better than the best value found. Lastly, this step terminates when the process hits the total number of iterations as well as the maximum number of non-improving iterations.

2. Vehicle scheduling: LP model is constructed.

The exact evaluation calculates the total travel time, the objective function of LP and the cost of used vehicles. Additionally, a self-adjusted demand infeasibility term is added during the neighbor search. We proposed three approximation methods and these methods were compared with those of exact evaluation. The outcome of this test can be interpreted as the quality criteria of the different approximation methods. Those methods are also compared among each other to determine the best performing one. Results show that the algorithm provides good quality solutions to our problem, while consuming reasonable computational efforts.

In a dynamic world, to address the real world problems effective and efficient decision support tools are needed. People from sales and logistics departments can benefit from the flexible version of the classical VRPTW which provide solutions by a faster heuristic during fleet planning and sales negotiation.

There are many perspectives that are worthy of receiving further investigation in future study. The more successful implementations of Tabu Search are more likely to create better initial solutions and neighborhood structures. Alternative strategies of generating an initial solution, more sophisticated neighborhood exploration, different memory structures, different aspiration criteria and more sophisticated diversification and intensification methods can be developed. One should also take the trade off between complexity of the algorithm and computational effort that this algorithm requires into consideration.

Table of Contents

Abstract	2
Preface.....	3
Executive Summary.....	4
Table of Contents.....	6
List of Figures	8
List of Tables.....	8
Table of Notations	9
1. Introduction.....	10
2. Literature Review.....	12
2.1. VRP	12
2.2. VRPTW.....	12
2.3. VRPFTW.....	14
2.4. Solution Approaches.....	14
3. Problem Analysis.....	16
3.1. Problem Definition.....	16
3.2. The VRPFTW Model	18
3.2.1. Decision Variables.....	18
3.2.2. The objective function	19
3.2.3. The VRPFTW Model	19
3.3. Methodology.....	20
4. Scheduling.....	21
5. Tabu Search for VRP with Flexible Time Windows.....	23
5.1. Initial Solution	23
5.2. Pseudo-code for Time-oriented, Nearest Neighbor Heuristic	24
5.3. Neighborhood Generation and Evaluation.....	25
5.3.1. Exact Evaluation	26
5.3.2. Cost of Demand Infeasibility	26
5.3.3. Approximate Evaluation.....	26

5.4. The Algorithm	28
6. Computational Experiments.....	30
6.1. Initial Parameter Setting	30
6.1.1. Parameter Setting for Time Oriented-NNH	31
6.2. Move Selection	31
6.2.1. Approximation Evaluation	31
6.2.2. Comparisons with the Exact Evaluation.....	32
6.3. Parameter Setting	34
7. Conclusion.....	36
References.....	38
Appendices.....	40
Appendix 1- Parameter Setting for Time Oriented NNH.....	40

List of Figures

Figure 1-Customer Satisfaction in VRPTW	16
Figure 2-Customer Satisfaction in VRPFTW	17
Figure 3-Lower and Upper Bound Violations for Time Window Violations.....	17

List of Tables

Table 1- Table of Notations.....	9
Table 2-Comparison of three approximation methods	31
Table 3-Comparison with Benchmark.....	33
Table 4-Defining Value Sets for Parameters	34
Table 5-Results for selected Solomon sets with different parameter settings	35

Table of Notations

Table 1- Table of Notations

Notation	Definition
K	Number of identical vehicles
C	Capacity of each identical vehicle
N	Number of customers
c	The cost of travelling one unit of distance
c_k	The cost of activating the vehicle $k \in K$
e_i	Earliest service start time of customer i
l_i	Latest service start time of customer i
$P_{max\ i}$	The maximum allowance for violation of time windows of customer i
c_{ei}	The unit penalty for the service that begins before its earliest start time
c_{li}	The unit penalty for the service that begins after its latest start time
q_i	The demand of customer i
d_{ij}	The distance between two different customers i and j
t_{ij}	The travel time between two different customers i and j
u_i	The service time for loading/unloading activities at customer i
a_i	The arrival time of the vehicle at customer i
s_i	The service start time at customer i
δ_i	The non-negative weight of the terms in cost metric of time-oriented NNH
φ	Unit penalty cost of demand infeasibility
κ	Tenure size
v_{ij}	The urgency of servicing the customer j after serving customer i
γ	Unit deviation cost of Approximation Method 2
δ	Unit deviation cost of Approximation Method 3

1.Introduction

Freight transportation is one of the most critical activities in supply chain management. This importance comes from the fact that it brings more than half of the total logistics cost. The contribution of the freight transportation cost to the total cost can be decreased by better utilization of the resources, which can be suggested by better routing and scheduling approaches to the problems. The Vehicle Routing Problem (VRP) thus has drawn a lot of attention. The VRP aims at designing a set of vehicle routes through several customer locations with minimum costs, under the conditions that each route starts and ends at the depot and each customer must be visited only once by one vehicle.

VRP can be described as a combination of two well-studied problems, namely Traveling Salesman Problem (TSP) and the Bin Packing Problem (BPP); since it conceptually lies at the intersection of those well-studied problems. TSP is the problem of finding one shortest possible tour that visits each city exactly once with a given list of cities and their pair wise distances. BPP is the minimization of the number of bins used to pack a given number of objects with different volumes and an instance of this kind of problem can be seen as a feasible solution for an instance of the VRP. Since TSP and BPP are considered as NP hard; computational effort required solving a VRP problem increases exponentially with the problem size; thus making it a NP-hard problem.

Some other constraints might need to be added to VRP depending on the problem setting, such as the time windows during which it is allowed to service the customers, which forms an important variant of the classic VRP – the Vehicle Routing Problem with Time Windows (VRPTW). The VRPTW is one of the critical problems that have been extensively studied by the researchers in the field. The VRPTW belongs to the class of the NP-hard combinatorial optimization problems. (Lenstra et al., 1981)

VRPTW involves the added complexity that every customer should be served within a given time interval, called “customer time window”. By adding that constraint, customers have enforced the use of time windows during the last few years in distribution, motivated by the Just-in-Time (JIT) principles and the competition arisen by adopting such principles. (Hopp et al., 1996) Time windows establish hard constraints to the delivery problem; thus forcing distribution companies and manufacturers to sustain their own vehicles and to increase their fleet size with the motivation of satisfying time window requirements.

The VRPTW has several practical applications in industries and services such as the distribution of cash amounts among bank branches, disposal of garbage and industrial wastes, distribution of fuel to and among fuel stations and school transportation services.

In this paper we consider the vehicle routing problem with flexible time window constraints (VRPFTW), in which vehicles are allowed to start servicing customers before and after the earliest and latest time window bounds, respectively. The time windows are often relaxed to allow for early or late arrivals at customer locations. That relaxation comes at the penalty costs as the time window violations has an effect on the customers' satisfaction. However, in applications where increasing customer satisfaction level is much more important, the penalty for earliness or tardiness needs to be minimized.

The problem setting proposed in this paper objects to provide an effective tool for time window adjustments which is also able to optimize the fleet size during the sales negotiation. The solution approach relies on relaxing the time windows and reducing the number of vehicles used compared to the fleet size in typical VRPTW instance while keeping time window violations to a bare minimum. The motivation is that by allowing limited time window violations for some customers, it may be possible to obtain significant reductions in the number of vehicles required and the total distance or time of all routes.

Each problem is solved using a collaborative hybrid algorithm, a combination of Tabu search and a linear programming problem. The actual evaluation of the target function is obtained by solving the resulting linear model to optimality for each route separately after a Tabu search heuristic for assigning customers to routes and for the sequencing of each route is used.

The remainder of this report is arranged as follows. In chapter 2, a brief literature review of related topic is provided. In chapter 3, the problem of the project is defined and described. The VRPFTW heuristic is proposed in chapter 5; after the linear model for scheduling within a given route is discussed in chapter 4. In chapter 6, the experiments and the results are presented and analyzed. And finally, the discussion and the conclusion are the discussed in the last chapter.

2.Literature Review

In this chapter the literature review of related topics is presented. The concept of VRP and VRPTW are presented with different approaches used in literature to tackle the problem.

2.1. VRP

Vehicle Routing Problem (VRP) is defined as the problem of designing optimal delivery or routes from one or several depots to a number of geographically scattered cities or customers, subject to side constraints. (Laporte, 1992) Due to the introduction of wide variety of constraints in terms of capacity, route length, time windows and precedence relations between customers; it is pretty hard to find a generally accepted definition of VRP. (Laporte, 2007).

VRP can be described as a combination of two well-studied problems, namely Traveling Salesman Problem (TSP) and the Bin Packing Problem (BPP); since it conceptually lies at the intersection of two well-studied problems. As previously studied, (Gendreau et al., 1994) solutions of the VRP are sometimes transformed into the TSP by replication of the depot.

Including also concept of TSP, VRP falls into category of NP-hardness. The most sophisticated exact algorithms for the VRP are able to solve only instances of up to around 100 customers with different succession (Baldacci et al., 2008). That's mostly why; researchers are forced to put an effort on heuristic algorithms in addition to fact that they give flexibility of dealing with the diversity of variants arising in practice.

2.2. VRPTW

The vehicle routing problem with time windows is the problem of designing least cost routes from one depot to a set of geographically scattered points in a way that each point visited only once by exactly one vehicle within a given time interval. All routes start and end at the depot and the total demand of all points on a route must not exceed the capacity of the vehicle. (Braysy,O. and M. Gendreau, 2005a). In other words, time windows requirement of each customer needs to be satisfied.

That variant of VRP, VRPTW, is also NP-hard. Savelsbergh (1985) studied that even getting a feasible solution to VRPTW when the fleet size is fixed is an NP-hard problem. Therefore, heuristics are again the most possible solution approach.

Customer time window can be characterized by an early time and a late time within which service should begin. Time windows concept exists in many real life situations such as dial-a-ride services, school-bus routes and bank deliveries. The planning needs to be designed in a way to consider issues of personnel availability to load the vehicles, traffic regulations and conditions, and some other predefined customer preferences. Interested reader is referred to Baker et al. (1986) and Solomon (1987).

Solomon (1987) has studied a number of heuristics including "savings algorithm", "time-oriented nearest neighbor algorithm", "insertion algorithm" and "time-oriented sweep algorithm" for VRPTW and additionally constructed a set of benchmark problems for an easy comparison of wide-variety of solution procedures. Numerous researchers have been using the Solomon test-sets when they test their heuristics or exact algorithms.

Van Landeghem (1988) presented his bi-criteria heuristic based on the Savings heuristic for VRPTW and discussed that the interaction between spatial and temporal issues complicates to understand the underlying dynamics of the problem.

Kolen et al. (1987) introduced an optimal algorithm based on branch-and-bound concept for VRPTW but failed to implement it in large problem settings. The algorithm is able to solve problems up to four vehicles serving fourteen customers.

Another algorithm based branch-and-bound is developed by Desrochers et al.(1992). The LP relaxation of the model is solved by column generation and a shortest path problem with time windows and capacity constraints is solved to add feasible columns as needed. That approach has been successful in optimally solving the problems with narrow time windows.

One of the algorithms that Fisher et al. (1991) has developed is based on a k-tree relaxation with time-windows as side-constraints and alternatively, the other algorithm introduces a solution for a semi-assignment problem and then treats the problem as a shortest path problem with time windows and capacity requirements. The researchers have tested these algorithms with benchmark problem sets and found optimal solutions.

The interested reader is referred to surveys of the published work in the field of VRP and VRPTW composed by Golden et al. (1988), Gendreau (1997) et al. and Laporte (1992) concerning solution methods developed for these problems in the last twenty five years.

2.3. VRPFTW

Customer time window is bounded by the earliest and latest time of the day that the delivery to the customer has to take place. In vehicle routing problem with flexible time windows, time windows can be violated by employing the appropriate penalties to reflect the negative effects of customer satisfaction. Violation is defined as the early or late arrival to the particular customer location at a cost of a penalty proportional to the extension in the time window.

That problem variant can be seen as a relaxation of the VRPTW and the literature reveals very little published work; in contrast to its applicability in practical cases. Below, we examine the limited research efforts for the VRPFTW.

Dumas et al. (1992) presented a procedure for selecting the time period at which each customer must be served to minimize the total inconvenience costs. Moreover, Ferland et al. (1989) introduced an algorithm that aims at adjusting time windows of pair of customers to reduce the total costs. Koskosidis et al. (1992) have designed an algorithm at which customers are assigned to vehicles using the general assignment method and then the TSP with time windows is solved separately for each vehicle.

Balakrishnan (1993) has created a linear penalty function for each customer to allowable limits of the earliest and latest service start times. The proposed model has been solved with different algorithms from nearest neighbor to penalty-expanded savings algorithm which obtained low-cost schedules serviced by fewer vehicles compared to no-violations allowed, typical VRPTW case.

2.4. Solution Approaches

Several exact and approximate algorithms have been proposed to get the solutions of VRP and its variants. NP-hardness of those kind of problem settings requires heuristic solution strategies for most real-life instances. Exact algorithms, based on branch-and-bound techniques, are very satisfying for only relatively small problems; but it is also studied that a number of approximate algorithms have provided promising results for larger problems. Those approximate algorithms include classical heuristics and metaheuristics.

Classical heuristics has two different types; namely constructive heuristics and improvement heuristics. The mostly used constructive heuristics is the Clarke and Wright savings algorithm (Clarke, G., and J.V.

Wright, 1964), which is initiated by placing the customer itself and the depot in one vehicle and then linking those vehicles according to the savings obtained while keeping the same requirements in several iterations. Other important classical heuristics include, e.g., petal heuristics, the sweep algorithm (Gillett, B.E., and L.R. Miller, 1974), a heuristic based on a two-phase decomposition procedure (Fisher, M.L., and R. Jaikumar, 1981). However, the performance of classical improvement heuristics is often not satisfying; thus used as building blocks within metaheuristics.

Improvement heuristics can be divided into intra-route and inter-route heuristics. Intra-route heuristics optimize each route alone with the help of a TSP improvement heuristic; whereas inter-route heuristics consist of moving vertices to different routes. (Laporte, G., and F. Semet, 2002).

Although metaheuristics needs more computation time than classical heuristics, but given the improvements in solution quality, the extra computational effort is well justified. (Gendreau, M., A. Hertz, and G. Laporte, 1994) Metaheuristics can be divided into three categories: local search, population search, and learning mechanisms. Tabu Search (TS) is one of the mostly used local search methods for VRPTW and its variants. Tabu Search will be explained in detail and applied to the first phase of the VRPFTW in Chapter 5.

3. Problem Analysis

In this chapter, the problem environment in our context is provided with notations, assumptions, the objective function and the relative constraints. Then, methodologies of the specific implementation are proposed.

3.1. Problem Definition

Consider a set of K identical vehicles and with a known capacity C , servicing a set of N customers originating from and terminating at the depot aiming at minimum cost.

The problem can be stated as follows. Let $G = (V, A)$ be a complete directed graph with $V = \{0, 1, \dots, N\}$ the set of vertices and A the set of links. The vertices represent the customers where V_0 is the depot. The non-stochastic travel time from customer i to customer j is represented by t_{ij} ; whereas the distance between customers i and j is associated with d_{ij} . The cost of travelling one unit of distance is c . Furthermore, there is a one-time cost of c_K for activating the vehicle $k \in K$.

Each customer i also has a standard service time u_i for loading and unloading activities. The service time for the depot is set to zero.

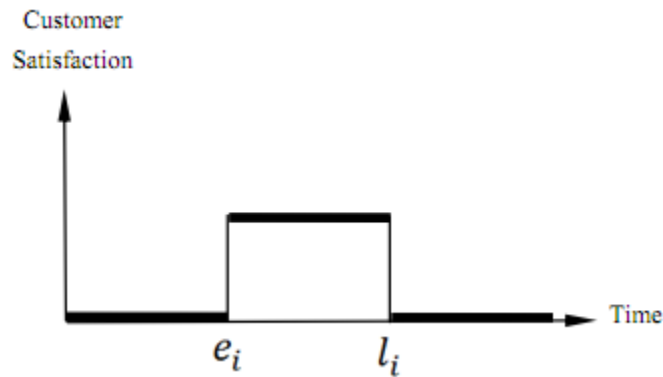


Figure 1-Customer Satisfaction in VRPTW

Each customer i possesses demand of q_i units and beginning of a service time denoted by s_i . A vehicle must start servicing the customer i between time intervals $[e_i, l_i]$. The customer time windows are often relaxed to allow for early or late arrivals at customer locations. That relaxation comes at the penalty costs as the time window violations has a effect on the customers' satisfaction, as showed in Figure 1 and Figure 2.

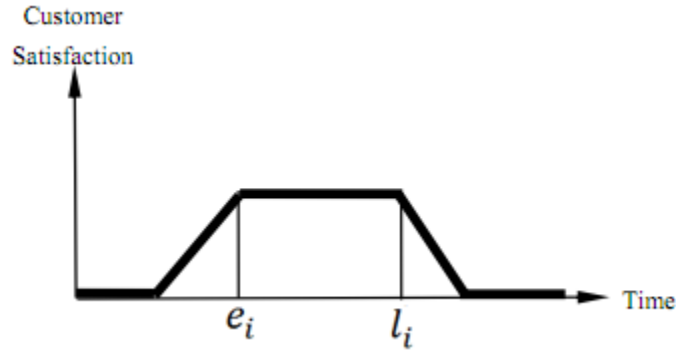


Figure 2-Customer Satisfaction in VRPFTW

Note that this allowance for violation of time windows is denoted by $P_{MAX} \geq 0$. That means time windows of each customer can be extended to $[e_i - P_{MAX}, l_i + P_{MAX}]$ (See Figure 3.) That extension is denoted by $pmaxc$ percent of the customer time window width and needs to be penalized proportional to lateness. For each customer, let c_{ei} be the unit penalty for the service begins before its earliest start time and c_{li} be the unit penalty for the service begins after its latest start time.

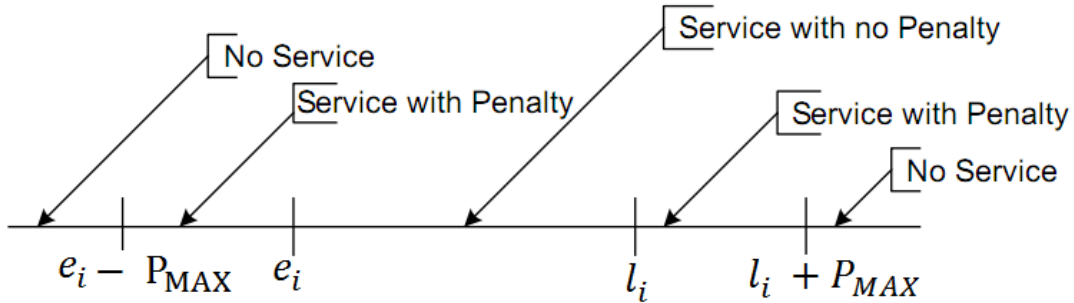


Figure 3-Lower and Upper Bound Violations for Time Window Violations

The formula for penalty function Y_i is as follows:

$$Y_i = \begin{cases} \infty & \text{if } s_i < e_i - P_{max} \\ c_{ei} (e_i - s_i) & \text{if } e_i - P_{max} \leq s_i \leq e_i \\ 0 & \text{if } e_i \leq s_i \leq l_i \\ c_{li} (s_i - l_i) & \text{if } l_i \leq s_i \leq l_i + P_{max} \\ \infty & \text{if } l_i + P_{max} < s_i \end{cases}$$

Assumptions

- I. The travel time between two vertices is non-stochastic, undirected and is proportional to travel distances. Furthermore, the triangle inequality is satisfied for the travel times.
- II. All vehicles are identical.
- III. The allowance for violation of time windows is denoted by $P_{MAX} \geq 0$.
- IV. The service time for the depot is set to zero.

3.2. The VRPFTW Model

3.2.1. Decision Variables

The model formulation requires three groups of variables:

- a. The first group of variables is binary and determines the sequence that vehicles visit customers:

$$X_{ij}^k = \begin{cases} 1 & \text{if customer } i \text{ is followed by customer } j \text{ in the sequence visited by } k \\ 0 & \text{otherwise} \end{cases}$$

- b. The second group of variables is also binary and checks if the vehicle is active:

$$z_k = \begin{cases} 1 & \text{if vehicle } k \text{ is active} \\ 0 & \text{otherwise} \end{cases}$$

- c. The third group determines the service start time of each customer and is represented with s_i .

3.2.2. The objective function

Given the above definitions, variables and parameters; we can introduce the objective function for the problem which should include three parts: travel cost, vehicle activation cost and tardiness penalty.

$$\sum_{k=1}^K \sum_{i=1}^N \sum_{j=1}^N c d_{ij} x_{ij}^k + \sum_{k=1}^K c_k z_k + \sum_{i=1}^N (c_{ei} \Delta_i^e + c_{li} \Delta_i^l)$$

In this expression, Δ_i^e represents the deviation from the time windows due to earliness and Δ_i^l represents the deviation from the time windows due to tardiness.

$$\Delta_i^e = \max\{0; e_i - s_i\}$$

$$\Delta_i^l = \max\{0; s_i - l_i\}$$

3.2.3. The VRPFTW Model

$$\sum_{k=1}^K \sum_{i=1}^N \sum_{j=1}^N c d_{ij} x_{ij}^k + \sum_{k=1}^K c_k z_k + \sum_{i=1}^N (c_{ei} \Delta_i^e + c_{li} \Delta_i^l)$$

subject to

$$\sum_{i=1}^N x_{i0}^k = 1 \quad \forall k \quad (1)$$

$$\sum_{i=1}^N x_{0i}^k = 1 \quad \forall k \quad (2)$$

$$\sum_{i=0}^N \sum_{k=1}^K x_{ij}^k = 1 \quad \forall j \quad (3)$$

$$\sum_{j=0}^N \sum_{k=1}^K x_{ij}^k = 1 \quad \forall i \quad (4)$$

$$\sum_{i=0}^N x_{im}^k - \sum_{j=0}^N x_{mj}^k = 0 \quad \forall k, m \quad (5)$$

$$\sum_{i=0}^N q_i \sum_{j=0}^N x_{ij}^k \leq C \quad \forall k \quad (6)$$

$$s_i \geq e_i - P_{max} \quad \forall i \quad (7)$$

$$s_i \leq l_i + P_{max} \quad \forall i \quad (8)$$

$$z_k \geq \sum_{i=1}^N x_{0i}^k \quad \forall k \quad (9)$$

$$s_i + u_i + t_{ij} - s_j \leq (1 - x_{ij}^k) M \quad \forall i, j, k \quad (11)$$

$$\Delta_i^e \geq e_i - s_i \quad \forall i \quad (12)$$

$$\Delta_i^l \geq s_i - l_i \quad \forall i \quad (13)$$

$$x_{ij}^k, z_k \in \{0,1\} \quad (14)$$

$$\Delta_i^e, \Delta_i^l \geq 0 \quad (15)$$

Constraints **(1)** and **(2)** make sure that each route starts and terminates at the depot, in other words at customer zero. Constraints from **(3)** to **(5)** assure that exactly one vehicle enters, serves and leaves each customer. Constraint **(6)** indicates that vehicle capacity is not exceeded. Constraints **(7)** and **(8)** determine the lower and upper boundaries for extended service start time of each customer. To specify the used vehicles, constraints **(9)** and **(10)** are needed. Constraint **(11)** ensures that if the vehicle travels from i to j , service at j cannot start earlier than that at i . Here, M is a very large constant. As explained earlier in the report, constraints **(12)** and **(13)** determine the tardiness that will be penalized in the objective function.

3.3. Methodology

The solution method for VRPFTW proceeds in two stages: the route planning and the service scheduling. For the route planning, the initial solution is formed by some specific route construction method. Afterwards, a Tabu Search algorithm is used for a certain iterations to pursue the improvements of the routes. The assignment of customers to vehicles and sequencing of customers are done via Tabu Search. In the second stage, the service scheduling is to be made via an LP model by using the solution obtained in the first phase. The decisions are made to minimize the total costs of the given solution. The derived LP model for this problem is developed in Chapter 4.

4. Scheduling

There will be a set of predetermined routes $Z = \{R_1, R_2, R_3, \dots, R_{|Z|}\}$ with $|Z| \leq K$. Each route $R_r = (0, i, j, \dots, 0)$ with n_r number of elements is formed by $\{0, 1, \dots, N\}$. The problem here is to find an optimal schedule for a given route R_r , at minimum cost. In other words, an LP model aims at minimizing the deviations from the given customer time windows.

Let a_j represent the arrival time at customer j . When the vehicle leaves a customer point (i.e. i), the time that it starts servicing (s_j) the next customer (i.e. j) in the sequence is determined by one of the three feasible scenarios:

1. If the vehicle arrives within the boundaries of time windows allowed for that customer point, it immediately starts servicing. That is, $a_j \in [e_j, l_j]$ and $s_j = a_j$.
2. If the vehicle arrives at the customer point later than latest service start time (l_j) or earlier than earliest service start time (e_j) of the customer thus leading to a violation with the maximum value of P_{max} , it immediately starts servicing in order not to pay more penalty cost. That is, $a_j \in [l_j, l_j + P_{maxj}]$ or $a_j \in [e_j - P_{maxj}, e_j]$ and $s_j = a_j$.
3. If the vehicle arrives at the customer point even earlier than its possible earliest service start time ($e_j - P_{max}$) of the customer, it needs to wait until the feasible lower boundary of the time windows, start servicing and pay the penalty. That is, $a_j \in [-\infty, e_j - P_{maxj}]$ and $s_j = \max\{e_j - P_{maxj}, a_j\}$

Minimize

$$\sum_{i=1}^{n_r} (c_{ei}\Delta_i^e + c_{li}\Delta_i^l)$$

subject to

$$s_i + u_i + t_{ij} \leq s_j \quad \forall i \in R_r \quad (1)$$

$$s_i \geq e_i - P_{max} \quad \forall i \in R_r \quad (2)$$

$$s_i \leq l_i + P_{max} \quad \forall i \in R_r \quad (3)$$

$$\Delta_i^e \geq e_i - s_i \quad \forall i \in R_r \quad (4)$$

$$\Delta_i^l \geq s_i - l_i \quad \forall i \in R_r \quad (5)$$

$$\Delta_i^e, \Delta_i^l \geq 0 \quad \forall i \quad (6)$$

In this LP, the only set of decision variables determines the service start time of each customer and is represented with s_i . The objective is to minimize the penalty cost subject to constraints listed from (1) to (6). Constraint (1) ensures that if the vehicle travels from i to j , service at j cannot start earlier than that at i . Note that, the travel time from i to next customer in the given sequence, j , is represented by t_i . Constraints (2) and (3) determine the lower and upper boundaries for extended service start time of each customer. Constraints (4) and (5) determine the tardiness that will be penalized in the objective function. Constraint (6) makes sure that the tardiness cannot be negative.

5. Tabu Search for VRP with Flexible Time Windows

The tabu search procedure generates a set of routes that still need to be scheduled using the LP previously described and determines the active vehicles, the number of routes, customers in each vehicle and the sequence that the vehicle visits them. The overall procedure is described in pseudo-code as in Section 5.3.

5.1. Initial Solution

In this project we use a fast and easy constructive algorithm “time-oriented, nearest neighbor heuristic” (Solomon,1985) for the initial solution, such that it starts every route from the depot, by each time finding the closest unvisited customer as long as all the restrictions (time windows, vehicle arrival time and capacity) are met, and then starts another tour.

The procedure of inserting a new customer is repeated until no other non-routed customer can be inserted into the route under construction. At this point, a new route is initialized with a different customer and the same procedure is executed until all customers are assigned to routes. Figure 4 presents the procedure of the construction of one route.

We will use a cost metric C_{ij} that measures the geographical and temporal closeness of the customers between customers i and j . The heuristic selects customer j with the lowest cost C_{ij} for the inclusion after the customer i . Note that, the criteria is sequence dependent; thus the relation between the last customer added to the route (i.e. i) and the new customer in the route (i.e. j) is taken into consideration.

That criterion makes sure that the selected customer j will be the closest to last routed customer i in terms of decisions on time windows violation and time influence.

In the equation of cost metric below, the weights $\delta_i, i \in \{1,2,3,4\}$ define the contribution of each individual metric to the overall criteria.

$$C_{ij} = \delta_1 t_{ij} + \delta_2 T_{ij} + \delta_3 v_{ij} + \delta_4 Y_j$$

t_{ij} : The direct travel time between two customers.

$T_{ij} = s_j - (s_i + u_i)$: The time between the beginning of service at j . It involves the time to travel the next customer point and the waiting time for the cases that have an arrival before the feasible service start time. (i.e. when $a_j \in [-\infty, e_j - P_{\max j}]$)

$v_{ij} = \max \{ 0, l_j - (s_i + u_i + t_{ij}) \}$: The urgency of servicing the customer j . It is defined as the time left until the latest service start time of customer j . This term represents the influence regarding the order of time-windows of customers on the shipping route. If we also included the amount of allowed violation $P_{\max j}$ into the term, we would have incurred the deviation from the time windows twice; both within the penalty cost and urgency term.

Y_j : The cost metric will include the penalty cost component, Y_j as defined below in addition to elements in Solomon's proposition. Here, the penalty component also controls the lower and upper bounds for the time windows violation ($P_{\max j}$) by stating the intervals for the service start time as follows:

$$Y_j = \begin{cases} \infty & \text{if } s_j < e_j - P_{\max j} \\ c_{ej}(e_j - s_j) & \text{if } e_j - P_{\max j} \leq s_j \leq e_j \\ 0 & \text{if } e_j \leq s_j \leq l_j \\ c_{lj}(s_j - l_j) & \text{if } l_j \leq s_j \leq l_j + P_{\max j} \\ \infty & \text{if } l_j + P_{\max j} \leq s_j \end{cases}$$

The non-negative weights represented with $\delta_1, \delta_2, \delta_3, \delta_4$ should satisfy $\delta_1 + \delta_2 + \delta_3 + \delta_4 = 1$ and $\delta_1 \geq 0, \delta_2 \geq 0, \delta_3 \geq 0, \delta_4 \geq 0$.

5.2. Pseudo-code for Time-oriented, Nearest Neighbor Heuristic

WHILE (there is an unvisited customer)

Find the "closest" customer to the previously routed customer according to the criteria C_{ij}

IF the closest can be served within its TW

Get the demand of the closest

```

IF total demand > capacity

    Find the next TW and capacity constraints satisfying unvisited closest

ELSE

    Include the closest into the route

END IF

ELSE

    Initialize a new route

END IF

END LOOP

```

5.3. Neighborhood Generation and Evaluation

The neighborhood of a solution contains all solutions that can be reached by moving nodes with some neighborhood generation methods. Several neighborhood generation methods are available in literature, including both intra-route exchange operators (e.g., 2-opt, Or-opt) and inter-route exchange operators (e.g., 2-opt*) In this project, we construct 2-opt* and Or-opt neighborhoods for the n nodes closest to i .

Evaluation of each neighborhood solution needs a separate LP-run. However, such computation is costly when a large problem size is performed to optimality. Thus, we have used approximate procedures that give an optimal or near optimal solution in tolerable time. We have proposed two different options in selecting the best move in its current neighborhood. In this project the one with smallest evaluation value is selected as the current solution.

After the move is selected, its exact evaluation is done by running the LP model for the changed routes to get an optimal schedule.

We describe the two criteria that allow avoiding the LP model for each candidate solution and leading to computationally efficient move selection process.

5.3.1. Exact Evaluation

Let's call the exact evaluation function of any solution S (feasible or not) $F(S)$. If we express the objective function in the LP model of a solution S (infeasible or not) $F_0(S)$, then

$$F_0(S) = \sum_{r \in R_r} \left\{ \sum_{i=1}^{n_r} (c_{ei} \Delta_i^e + c_{li} \Delta_i^l) \right\} = \sum_{R_r \in S} \Theta(R_r)$$

For any feasible solution S , the total cost function is,

$$F_T(S) = c \sum_{R_r \in S} \sum_{(i,j) \in R_r} d_{ij}$$

$$F_c(S) = F_0(S) + F_T(S) + \sum_{k=1}^K c_k Z_k$$

where $F_T(S)$ calculates the total travel cost of the solution and the third part calculates the cost of used vehicles in the solution.

5.3.2. Cost of Demand Infeasibility

As Gendreau et al.(1994) proposes that allowing the existence of routes with total demand exceeding the vehicle capacity, in other words allowing "demand infeasible solutions", and penalizing such solutions proportionally to their capacity brings diversified search to the solution. Thus, for any infeasible solution S , a penalty term should be added and the evaluation function is replaced by the function:

$$F_{TT}(S) = F_T(S) + \varphi \sum_{R_r \in S} \left[\left(\sum_{i \in R_r} q_i \right) - Q \right]^+$$

In this new formula, the term $\varphi \sum_{R_r \in S} [(\sum_{i \in R_r} q_i) - Q]^+$ penalizes the demand infeasible route.

5.3.3. Approximate Evaluation

Below, the three criteria that allow avoiding the use of the LP model for each candidate solution and lead to computationally efficient move selection procedures are proposed:

Approximation 1 – Distance based

The heuristic is based on minimizing the modified travel cost $F_{TT}(S)$. It does not take customer time windows and the penalty cost function into account. Let S' denote the neighbor of the current solution S . Thus the heuristic selects the move that is not tabu and maximizes the following formula:

$$\Delta_1(S') = [F_{TT}(S) - F_{TT}(S')]$$

Approximation 2 – Distance based and the penalties of moves

As mentioned in Section 3.1 a customer time window can be relaxed to allow for early or late arrivals at customer locations. We let dev_i denote the deviation from the customer windows $[e_i, l_i]$ and dev_j denote the deviation from the customer windows $[e_j, l_j]$. Those deviations represent the minimum amount at customer locations to make the route feasible by adding flexibility.

Each deviation unit is penalized by γ . For each solution S' involving a move between customer i and customer j , we compute the following quantity

$$\Delta_2(S') = [F_{TT}(S) - F_{TT}(S')] - \gamma[dev_i + dev_j]$$

where $dev_i = \max\{e_i - s_i, 0, s_i - l_i\}$.

This heuristic selects the move that is not tabu and maximizes the formula above. The logic behind the formula is based on the elimination of the moves that require higher deviations are likely to decrease the total cost associated with the route.

Approximation 3 – Distance based and the penalties for the entire route

That heuristic is based on the same concept with Approximation 2; penalizing the deviations. It differs in a way that it does not only consider the deviations by the changed moves but also the deviations occurred within the sub-route; from the first move in the route to the last customer in the route. i.e. a push or a pull movement of the service start time of a certain customer, after the move involving that customer is executed, is able to affect the remaining customers in the route.

Consider a move between customer i from route R_1 and customer j from route R_2 , leading to the solution S' . Let n_1 denote the number of customer locations visited by the route R_1 and n_2 denote the number of customer locations visited by the route R_2 . Let $dev_1^{S'}$ represent the maximum of all deviations occurred within the route R_1 and $dev_2^{S'}$ represent the maximum of all deviations occurred within the route R_2 . The heuristic selects the move that is not tabu and maximizes the following formula:

$$\Delta_3(S') = [F_{TT}(S) - F_{TT}(S')] - \delta[dev_1^{S'} + dev_2^{S'}]$$

where $dev_1^{S'} = \max\{dev_i, dev_{i+1}, \dots, dev_{n_1+1}\}$ and $dev_2^{S'} = \max\{dev_j, dev_{j+1}, \dots, dev_{n_2+1}\}$

To sum up, we have used those three approximation methods to assess the possible moves based on the penalty values in the problem.

5.4. The Algorithm

Step 0: Initialization

Read $C, t_{ij}, c_K, c, q_i, e_i, l_i, c_{ei}, c_{li}, u_i, pmaxc, \delta_1, \delta_2, \delta_3, \delta_4$

Construct initial solution S_0 and compute $F(S_0)$

Set $S = S_0$ and $F(S) = F(S_0)$

Step 1: Neighborhood Generation and Evaluation

Generate the neighborhood of solution S

Evaluate each neighborhood solution by one of the three approximation methods and retain the best non-tabu move as new solution S

Evaluate $F(S)$ and update the tabu list to include S

Step 2: Improvements

If S is better than the current best solution, update the best feasible solution to S

Update the excess demand penalty

If no improvement in η_{max} iterations then

Store the best solution

Else go to Step 1

Step 3: *Terminate*

Output the following: Number of routes, sequence of customers visited by each vehicle, number of violated time windows, total time (distance), total cost

6. Computational Experiments

Our approach has been tested on the classical data sets R1, C1 and RC1 of Solomon (1987). Each data set contains problems with 100 customers. The Cartesian coordinates of customers in the R1 problems are randomly generated from a uniform distribution, while the coordinates of customers in the problems in set C1 are clustered. Problems in the set RC1 contain semi-clustered customers, i.e., a combination of clustered and randomly (uniformly) distributed customers. The vehicle capacity is 200 units for all problem sets. The service times and the time windows for customers are given. For additional information concerning the data sets, the reader is referred to the Solomon's study (1987).

The implementation of the algorithm was coded using Visual C++ 2008 Express Edition, and evaluated with simplex algorithm in the application of Gurobi Optimizer 4.5.0. Experiments ran on an HP Compaq 8530w Mobile Workstation with an Intel® Core™2 Duo CPU 2.80 GHz and 4 GB of RAM, and an operating system of Windows Vista™ Enterprise.

6.1. Initial Parameter Setting

To run the first group of experiments, we will use the values provided in the Solomon data sets and the parameter values that are widely used by the researchers in the field. Among these parameters, the penalties associated with approximations 2 and 3 are chosen as $\gamma = 0.1$ and $\delta = 0.1$, respectively by doing some preliminary tests. Important to mention that these values do not necessarily fit all of the 29 instances but provide reasonable results.

We use the terms distance and travel time interchangeably since the travel cost c in $F_T(S)$ is set to one. For an instance with N nodes, for each customer the $\eta = \lceil 0.3 N \rceil$ closest customers are candidates for a move. The tenure size κ is set to 20. The infeasibility penalty φ is set to 10.

We consider allowable penalty $P_{\max i}$ equal to 10 % of the customer i 's time window; as this ratio has been widely used by the researchers in the field. (Balakrishnan, 1993) Furthermore, the penalty coefficients c_{ei} and c_{li} are set to 1 for each customer i as Kritikos et al. (2002) proposed to use in their study. The vehicle activation cost, c_K is 20 for each vehicle.

In section 6.3, we will touch more on how different values of unit penalty costs and allowance for time window violation ($P_{\max i}$) can reflect the objective function.

6.1.1. Parameter Setting for Time Oriented-NNH

To examine the effect of the various parameters of time-oriented nearest neighbor algorithm on the solution quality, we run experiments with different set of values of the parameter set $(\delta_1, \delta_2, \delta_3, \delta_4)$ for all of the 29 Solomon instances. The parameters used for this table $(\delta_1, \delta_2, \delta_3, \delta_4)$ are: PS1(0,0.1,0,0.9), PS2 (0.3,0.2,0.3,0.2), PS3 (0.3,0.1,0.1,0.5), PS4 (0.3,0.4,0.2,0.1), PS5 (0.1,0.7,0.1,0.1) and PS6 (0.5,0.2,0.1,0.2). The number of used vehicles is chosen as the criteria to evaluate the different value sets and the results are provided in Appendix A.

In Appendix A, we provide the number of vehicles needed in the system for every Solomon instance-parameter set combination. From the data of Appendix A, we cannot extract concrete conclusions concerning parameters $\delta_1, \delta_2, \delta_3$ and δ_4 . Even a complete examination of all the results we obtained during our experiments could not provide strong evidence on a consistently performing set of values for these parameters. Here, we accept to use the value set PS 6 (0.5,0.2,0.1,0.2) for the parameters $(\delta_1, \delta_2, \delta_3, \delta_4)$ since it used the least number of vehicles in 19 out of 29 instances.

6.2. Move Selection

6.2.1. Approximation Evaluation

Table 2 shows the results of implementations for Solomon data sets in which only one of the three approximation methods has been used.

Table 2-Comparison of three approximation methods

Problem	Objective Value		
	Approximation 1	Approximation 2	Approximation 3
R101	1512.45	1423.72	1393.72
R102	1397.67	1387.71	1381.22
R103	1441.25	1381.23	1372.45
R104	1295.19	1258.68	1273.84
R105	1291.49	1276.89	1280.21
R106	1324.65	1298.55	1280.55
R107	1292.44	1277.22	1263.48
R108	1239.18	1187.27	1180.65
R109	1248.32	1238.94	1223.3

R110	1237.49	1202.84	1216.02
R111	1301.38	1252.96	1231.83
R112	1222.9	1200.42	1216.49
C101	1143.76	1117.87	1121.83
C102	1145.29	1133.74	1107.48
C103	1116.71	1109.31	1116.61
C104	1136.63	1114.73	1117.57
C105	1181.87	1134.22	1146.13
C106	1174.28	1163.82	1146.29
C107	1168.89	1132.14	1122.46
C108	1116.83	1114.95	1100.24
C109	1204.66	1128.51	1132.52
RC101	1476.54	1463.03	1436.42
RC102	1423.87	1402.27	1407.85
RC103	1357.39	1395.28	1345
RC104	1361.82	1327.56	1326.93
RC105	1518.54	1446.87	1423.83
RC106	1409.13	1381.9	1374.28
RC107	1373.72	1332.04	1337.21
RC108	1385.71	1306.41	1316.3

In Table 2, the columns on the right hand side shows the objective function obtained by employing one of the three approximation methods. We have observed that approximation 3 outperforms the other approximation methods in 17 out of 29 problems; while approximation 2 does so 12 times.

6.2.2. Comparisons with the Exact Evaluation

Table 3 shows the solution quality of different approximation methods. Such solution quality is tested by comparing the results obtained in Table 2 with the optimal solutions solved by the exact algorithm discussed earlier in Chapter 5.2.1. The columns on the right side of the table (Ratio 1, Ratio 2 and Ratio 3) are the percentage of the ratio of (Approximation/Exact). For instance, one could claim that the

approximation 2 provides a result that is about 105% of the one that exact algorithm gives for the problem R101.

As provided below, some of the instances using our algorithm have obtained relatively larger gaps with the benchmark values (solution obtained by the exact algorithm). The ratios can be interpreted as the quality criteria of the different approximation methods.

Table 3-Comparison with the Exact Evaluation

Problem	Exact	Ratio 1(%)	Ratio 2(%)	Ratio 3(%)
R101	1355.567	111.5733	105.0277	102.8146
R102	1322.933	105.6493	104.8964	104.4059
R103	1320.933	109.1085	104.5647	103.9
R104	1198.233	108.0916	105.0446	106.3098
R105	1235.3	104.5487	103.3668	103.6356
R106	1222.3	108.3736	106.2382	104.7656
R107	1155.3	111.8705	110.5531	109.3638
R108	1120.3	110.6114	105.9779	105.3869
R109	1175.3	106.2129	105.4148	104.0841
R110	1155.3	107.1142	104.1149	105.2558
R111	1160.3	112.1589	107.9859	106.1648
R112	1135.3	107.716	105.7359	107.1514
C101	1042.833	109.6781	107.1955	107.5752
C102	1062.833	107.7582	106.6715	104.2007
C103	1047.833	106.5732	105.867	106.5637
C104	1042.833	108.9944	106.8944	107.1667
C105	1062.833	111.1999	106.7166	107.9069
C106	1067.833	109.9685	108.9889	107.3473
C107	1062.833	109.9787	106.5209	105.6102
C108	1042.833	107.0957	106.9155	105.5049
C109	1055.833	114.0957	106.8833	107.2631
RC101	1380.1	106.9879	106.009	104.0809
RC102	1340.1	106.251	104.6392	105.0556

RC103	1280.167	106.0323	108.9921	105.0644
RC104	1269.7	107.2553	104.557	104.5074
RC105	1340.6	113.2732	107.927	106.2084
RC106	1309.7	107.5918	105.5127	104.9309
RC107	1249.7	109.924	106.5888	107.0025
RC108	1260.7	109.9159	103.6258	104.4102

6.3. Parameter Setting

In this section, we will select the first three instances from the classical data sets R1, C1 and RC1 of Solomon to show the procedure of parameter tuning and how different values of penalty costs(c_{ei} and c_{li}) and allowance for time window violation ($P_{max i}$) can reflect the objective function. We have defined four different parameter settings for this purpose, from Set 1 to Set 4 as follows:

Table 4-Defining Value Sets for Parameters

		$P_{max i}$	
		5%	10%
Penalty Cost (c_{ei} c_{li})	(1,1)	Set 1	Set 2
	(2,2)	Set 3	Set 4

Table 5 shows the results for the four experimental settings after running the Tabu Search procedure with the exact evaluation. The values provided in the table are the obtained target function values. On average, the objective values under Set 1 are only 0.43% higher than those under Set 2. That can be explained by the increment in the number of vehicles. On average, the objective values of Set 3 are 0.94% higher than those of Set 2; because of the tighter customer time windows and the doubled penalty cost(c_{ei} c_{li}). If the penalty cost is doubled with 10% allowance for violation the customer time windows, as showed in Set 4, the objective values are 0.62% higher than those of Set 2, on average. We could conclude that the changes are not dramatic.

Table 5-Results for selected Solomon sets with different parameter settings

Problem	Set 1	Set2	Set3	Set4
R101	1888	1355.56	3085	2118
R102	1757	1322.93	2150	2167
R103	1784	1320.93	3002	2241
R104	1901	1198.23	2610	1935
R105	1760	1235.3	2120	2023
C101	1385	1042.83	2136	1579
C102	1652	1062.83	1933	1686
C103	1556	1047.83	1737	1718
C104	1633	1042.83	1800	1648
C105	1585	1062.83	2312	1712
RC101	2079	1380.1	2698	2325
RC102	1799	1340.1	2184	2261
RC103	1787	1280.16	2234	1965
RC104	1923	1269.7	2365	1965
RC105	1855	1340.6	2718	2054
Average	1756.26	1220.18	2338.93	1959.8

7. Conclusion

In this thesis, we have described a methodology to solve a special variant of VRP – VRPFTW. Vehicle Routing Problem with Flexible Time Windows (VRPFTW) in which vehicles are allowed to start servicing customers before and after the earliest and latest time window bounds, respectively. The time windows are relaxed to allow for early or late arrivals at customer locations. That relaxation comes at the penalty costs as the time window violations has an effect on the customers' satisfaction. Violation is defined as the early or late arrival to the particular customer location at a cost of a penalty proportional to the extension in the time window and must be penalized to reflect the negative effects of customer satisfaction.

Our solution approach is a hybrid algorithm in which routing and scheduling are incorporated in a sequence. The routing component is handled via a Tabu Search procedure, while solving an LP model provides a robust vehicle scheduling.

The solution engine of the method, our algorithm, which is based on the time-oriented nearest-neighbor heuristic developed to account for a penalty associated with time window violations, is applied on Solomon's problem sets. It provides instances where vehicles are allowed to service customers before or after their specified time windows. The problem is crucial for fleet planning and contract negotiations since it enables decision-makers to determine the best trade-off between time window expansion and number of required vehicles.

The approximation methods, we have developed to avoid running the computationally inefficient exact evaluation, were tested to be compared with the exact evaluation. Those methods are also compared among each other to determine the best performing one. Results show that the algorithm provides good quality solutions to our problem, while consuming reasonable computational efforts.

In a dynamic world, to address the real world problems effective and efficient decision support tools are needed. People from sales and logistics departments can benefit from the flexible version of the classical VRPTW which provide solutions by a faster heuristic during fleet planning and sales negotiation.

There are many perspectives that are worthy of receiving further investigation in future study. The more successful implementations of Tabu Search are more likely to create better initial solutions and neighborhood structures. Alternative strategies of generating an initial solution, more sophisticated neighborhood exploration, different memory structures, different aspiration criteria and more

sophisticated diversification and intensification methods can be developed. One should also take the trade off between complexity of the algorithm and computational effort that this algorithm requires into consideration. Another future option can focus on the setting where only a subset of customers has fixed time windows. A study on developing more sophisticated approximation methods and doing an extensive parameter tuning of these methods can be conducted.

References

- Baldacci, R., N. Christofides, and A. Mingozzi. (2008). An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming* , 351-385.
- Baker, E., and J. Schaefer. (1986). Solution improvement heuristics for the vehicle routing and scheduling problem with time window constraints. *American Journal of Mathematical and Management Sciences*, 6, 261-300.
- Balakrishnan,N.(1993). Simple heuristics for the vehicle routing problem with soft time windows. *Journal of the Operational Research Society*, 44(3),279–287.
- Braysy,O., and M.Gendreau.(2005a). Vehicle routing problem with time windows, part I: Route construction and local search algorithms. *Transportation Science*, 39 (1), 104-118.
- Clarke,G., and J.V. Wright. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research* , 12, 568-581.
- Desrochers, M., J. Desrosiers, and M. Solomon.(1992).A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research*, 40, 342-354.
- Dumas,Y., F. Soumis, and J. Desrosiers. (1990). Optimizing the schedule for a fixed vehicle path with convex inconvenience costs. *Transportation Science*, 24, 145-152.
- Ferland, J. A., and L. Fortin. (1989). Vehicle routing with sliding time-windows. *Eur. J. Opl Res.*, 38, 213-226.
- Fisher,M.L., and R. Jaikumar. (1981). A generalized assignment heuristic for the vehicle routing problem. *Networks* , 11, 109-124
- Fisher, M.L., K. Jornsten, and B.G.Madsen.(1991).Vehicle routing with time windows preliminary results. Research Report 4, IMSOR, Technical University of Denmark, DK 2800,Lyngby, Denmark.
- Gendreau,M., A. Hertz, and G. Laporte. (1994). A tabu search heuristic for the vehicle routing problem. *Management Science* , 40 (10), 1276-1290.

Gendreau,M., G. Laport, and J. Potvin. (1997).Vehicle routing: modern heuristics. Local Search in Combinatorial Optimization Edited by E. Aarts and JK Lenstra,John Wiley & Sons,New York.

Gillett,B.E., and L.R. Miller. (1974). A heuristic algorithm for the vehicle dispatch problem. Operations Research , 22, 340-349.

Golden,B., and A. Assad.(1988) Vehicle routing: Methods and Studies. Elsevier Science. Publishers, Amsterdam.

Hopp,W.J., and M.L.Spearman.(1996) Factory physics: Foundations of manufacturing management. Irwin, Chicago, IL.

Kolen,A., A. Rinnooy Kan,and H. Trienekens. (1987). Vehicle routing with time windows. Operations Research,35, 266-273.

Koskosidis,Y., W. Powell, and M. Solomon. (1992). An optimization-based heuristic for vehicle routing and scheduling with soft time window constraints. Transportation Science, 26, 69-85.

Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. European Journal of Operational Research, 345-358.

Laporte, G. (2007). What you should know about the vehicle routing problem. Naval Research Logistics, 811-819.

Lenstra,J., and A. Rinnooy Kan.(1981). Complexity of vehicle routing and scheduling problems. Networks, 11, 221–227.

Savelsbergh,M. (1985). Local search in routing problems with time windows. Operations research, 4, 285-305.

Solomon,M. (1987). Algorithms for the vehicle routing and scheduling problem with time window constraints. Operations research,35, 254-265.

Van Landeghem, H.R.G. (1988). A bi-criteria heuristic for the vehicle routing problem with time windows. European Journal of Operations Research,36, 217-226.

Appendices

Appendix 1- Parameter Setting for Time Oriented NNH

Problem	Number of Vehicles					
	PS 1	PS 2	PS 3	PS 4	PS 5	PS 6
R101	23	23	23	25	23	22
R102	23	21	20	20	22	20
R103	23	21	20	20	22	20
R104	15	12	11	12	13	14
R105	16	17	16	16	15	16
R106	17	15	15	14	14	15
R107	13	13	12	13	12	12
R108	13	12	12	11	10	10
R109	14	14	13	14	13	13
R110	13	14	11	12	11	12
R111	15	13	12	11	12	12
R112	11	12	11	11	10	11
C101	11	12	11	11	10	10
C102	11	12	11	11	11	11
C103	12	10	10	10	10	10
C104	11	10	10	10	10	10
C105	11	12	10	11	11	11
C106	11	12	11	12	11	11
C107	11	11	10	11	11	11
C108	10	12	11	11	11	10
C109	10	11	11	11	11	10
RC101	18	17	19	17	17	18
RC102	16	15	14	15	15	16
RC103	15	14	13	13	13	13
RC104	14	12	13	13	13	12
RC105	19	18	18	18	17	16

RC106	14	15	14	15	14	14
RC107	14	13	12	12	12	11
RC108	14	13	12	12	12	11