# THE CAPACITATED VEHICLE ROUTING PROBLEM WITH SIMULTANEOUS PICKUP-DELIVERY AND TIME WINDOWS IN THE SUSTAINABLE FOOD SUPPLY CHAINS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MAHDI FARSHCHI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
INDUSTRIAL ENGINEERING

JANUARY 2021

Approval of the thesis:

**THE CAPACITATED VEHICLE ROUTING PROBLEM WITH SIMULTANEOUS PICKUP-DELIVERY AND TIME WINDOWS IN THE SUSTAINABLE FOOD SUPPLY CHAINS**

submitted by **MAHDI FARSHCHI** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Industrial Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ____________________

Prof. Dr. Esra Karasakal
Head of Department, **Industrial Engineering** ____________________

Assoc. Prof. Dr. Sedef Meral
Supervisor, **Industrial Engineering**, **METU** ____________________

Prof. Dr. Ferda Can Çetinkaya
Co- Supervisor, **Industrial Engineering**, **Çankaya University** ____________________

**Examining Committee Members:**

Prof. Dr. Meral Azizoğlu
Industrial Engineering, METU ____________________

Assoc. Prof. Dr. Sedef Meral
Industrial Engineering, METU ____________________

Assist. Prof. Dr. Sakine Batun
Industrial Engineering, METU ____________________

Assist. Prof. Dr. Diclehan Tezcaner Öztürk
Industrial Engineering, Hacettepe University ____________________

Assist. Prof. Dr. Hakan Özaktaş
Industrial Engineering, Çankaya University ____________________

Date: 14.01.2021

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Last Name : Mahdi Farshchi

Signature :

## ABSTRACT

### THE CAPACITATED VEHICLE ROUTING PROBLEM WITH SIMULTANEOUS PICKUP-DELIVERY AND TIME WINDOWS IN THE SUSTAINABLE FOOD SUPPLY CHAINS

Farshchi, Mahdi
Doctor of Philosophy, Industrial Engineering
Supervisor : Assoc. Prof. Dr. Sedef Meral
Co- Supervisor : Prof. Dr. Ferda Can Çetinkaya

January 2021, 159 pages

The aim of our study is transportation planning in perishable food supply chains. We model this problem as the vehicle routing problem with simultaneous pickup and delivery, and soft time windows for perishable products (VRPSPD-STW-P).

The objective is to minimize the total costs, consisting of variable transportation costs, vehicle-related fixed costs, food quality degradation costs, and time-window violation costs. We formulate the problem as a mixed-integer linear programming model. Moreover, we propose heuristic and metaheuristic algorithms in the methodology to solve the VRPSPD-STW-P.

Our methodology comprises two phases: obtaining initial solutions in the first phase and improving these solutions by a genetic algorithm in the second phase. In these phases, we first employ a method based on the clustering of nodes for the vehicles and different routing heuristics to generate the best routes for the vehicles; that is, solving the problem as an *m*-traveling salesperson problem with simultaneous pickup and delivery (*m*-TSPSPD). Then the solution of the problem *m*-TSPSPD is checked for feasibility taking into account the vehicle capacities and tour lengths; the feasible routes are then the solutions for the vehicle routing problem with simultaneous pickup and delivery (VRPSPD). Finally, the VRPSPD solution is evaluated considering the time window and quality constraints (VRPSPD-STW-P). Our solution methodology yields promising solutions in much less computational time than the solutions generated by the exact solution procedures.

Keywords : Sustainable food supply chain, Vehicle routing problem, Simultaneous pickup and delivery, Soft time windows, Mixed integer linear programming, Genetic algorithm.

# ÖZ

## SÜRDÜRÜLEBİLİR GIDA TEDARİK ZİNCİRLERİNDE ZAMAN PENCERELİ VE EŞZAMANLI TOPLAMA-TESLİMATLI KAPASİTE KISITLI BİR ARAÇ ROTALAMA PROBLEMİ

Farshchi, Mahdi
Doktora, Endüstri Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Sedef Meral
Eş Tez Yöneticisi: Prof. Dr. Ferda Can Çetinkaya

Ocak 2021, 159 sayfa

Çalışmamızın amacı, bozulabilir gıda tedarik zincirlerinde ulaştırma planlamasıdır. Bu problem, bozulabilir ürünler için eşzamanlı toplama-teslimat yapan esnek zaman pencereli araç rotalama problemi (VRPSPD-STW-P) olarak modellenir.

Amaç, değişken ulaşım maliyetleri, araçla ilgili sabit maliyetler, gıda kalitesi bozulma maliyetleri ve zaman penceresi ihlâli maliyetlerinden oluşan toplam maliyetin en aza indirilmesidir. Problem karışık tamsayılı doğrusal programlama modeli olarak formüle edilir. Ayrıca VRPSPD-STW-P problemini çözmek için geliştirilen metodolojide sezgisel ve metaheuristik algoritmalar önerilir.

Geliştirilen metodoloji iki aşamadan oluşur: birinci aşamada olurlu başlangıç çözümleri elde edilir ve ikinci aşamada genetik bir algoritma ile bu çözümler geliştirilir. Bu aşamalarda, ilk önce araçlar için talep noktalarının kümelenmesine ve en iyi rotaların belirlenmesi için farklı rota belirleme algoritmalarına dayanan bir yöntem kullanılır; bu yöntem aslında ana problemi, eşzamanlı toplama-teslimat yapma amacı ile talep

noktalarını ziyaret eden birden-fazla-gezgin-satıcılı problem olarak (m-TSPSPD) çözmeye denk gelir. Daha sonra m-TSPSPD'nin çözümü, araç kapasiteleri ve rota uzunlukları dikkate alınarak olurluğu açısından kontrol edilir; olurlu bulunan bu uygulanabilir araç rotaları, eşzamanlı toplama-teslimatlı (VRPSPD) araç rotalama probleminin çözümünü oluşturur. Son olarak, VRPSPD çözümü, talep noktalarının zaman penceresi ve kalite kısıtlamaları dikkate alınarak değerlendirilir (VRPSPD-STW-P). Çözüm metodolojimiz, kesin-çözüm yaklaşımları ile elde edilen çözümler ile karşılaştırıldığında çok daha az hesaplama süresi içinde umut verici çözümler sunar.

Anahtar Kelimeler: Sürdürülebilir gıda tedarik zinciri, Araç rotalama problemi, Eşzamanlı toplama-teslimat, Esnek zaman pencereleri, Karışık tam sayılı doğrusal programlama, Genetik algoritma.

*To my family for their endless support and encouragement*

## ACKNOWLEDGMENTS

Foremost, I would like to thank my thesis supervisor Assoc. Prof. Dr. Sedef Meral for her guidance, support, and encouragement. She provided me the best advise at all stages of my studies. She is a modest professor, patient with an excellent level of science. During my studies, I used her scientific experiences, and she always helped me with educational advancement. Her office door was always open whenever I ran into a trouble spot or had a question about my thesis. I learned a lot from her, and it was a big honor for me to know and work with her.

I want to thank my thesis co-supervisor Prof. Dr. Ferda Can Çetinkaya for his guidance, support, and encouragement. He is a professor with excellent scientific level and innovative ideas about industrial engineering. He is a modest and well-mannered professor.

I also wish to appreciate my thesis committee members Assist. Prof. Dr. Sakine Batun and Assist. Prof. Dr. Diclehan Tezcaner Öztürk for their invaluable feedback.

I am proud to have studied at the Middle East Technical University and also to have met excellent scientific level professors.

I want to thank my friends for sharing their experiences and memories with me, especially Amin Faridyahyaei for his help.

Finally, and most importantly, I declare my most appreciation to my family for their endless love, support, and encouragement.

# TABLE OF CONTENTS

## LIST OF TABLES

TABLES

## LIST OF FIGURES

FIGURES

# LIST OF ABBREVIATIONS

ABBREVIATIONS

| | |
|---|---|
| APD | Average percent deviation |
| APG | Average percent gap |
| API | Average percent improvement |
| GA | Genetic algorithm |
| ICR-CN | Independent clustering and routing- closest neighbor |
| ICR-TW | Independent clustering and routing - time window |
| MILP | Mixed-integer linear programming |
| RCR | Random clustering and routing |
| SCM | Supply chain management |
| SSCM | Sustainable supply chain management |
| STW | Soft time window |
| TW | Time window |
| TSP | Traveling salesperson problem |
| TSPSPD | Traveling salesperson problem with simultaneous pickup and delivery |
| VRP | Vehicle routing problem |
| VRPB | Vehicle routing problem with backhauls |
| VRPCB | Vehicle routing problem with clustered backhauls |
| VRPDPD | Vehicle routing problem with divisible pickup and delivery |
| VRPMB | Vehicle routing problem with mixed linehauls and backhauls |
| VRPPD | Vehicle routing problem with pickup and delivery |
| VRPSPD | Vehicle routing problem with simultaneous pickup and delivery |
| VRPSPD-STW-P | Vehicle routing problem with simultaneous pickup and delivery, and soft time windows for perishable products |
| VRPTW | Vehicle routing problem with time windows |

# CHAPTER 1

# INTRODUCTION

*Supply Chain Management* (SCM) is defined as the management of information, material flows, and cooperation among organizations. Traditional SCM comprises planning, sourcing, production, and distribution logistics and focuses on economic and financial business performance.

Global warming and depletion of natural, non-renewable resources and increasing industrial activities force industries to focus on *Sustainable Supply Chain Management* (SSCM). SSCM is an integration of environmental, social, and economic aspects as well. Companies are using different sustainable operations that improve some combination of environmental, social, and economic outcomes in different phases of the product life cycle, such as forward channel, reverse channel, pickup and delivery policies, production planning, and remanufacturing or product recovery.

Forward and reverse logistics play the primary role in environmental issues such as carbon emission and footprints, and food miles of a sustainable supply chain. Today's customers want to purchase sustainable products, and they are willing to have information about procurement processes, production and distribution methods. SSCM focuses on the forward supply chain, reverse logistics, remanufacturing, and product recovery.

Over the past years, sustainability has gained ever-increasing importance in the food industry. Making customers have environmentally responsible choices in their consumption, identifying the critical areas for environmental improvements, and exploring the status of information systems to support sustainability play a significant role in SSCM.

Besides the commonly used cost-based performance measures related to safety and quality, sustainability includes environmental and social dimensions. The environmental dimension of sustainability has probably received the most attention. Life cycle assessment (LCA) is one of the best-known examples, which is an analytical tool that helps in assessing a product's environmental impact from product development to consumption, i.e., the wasted product is an important performance measure in sustainability.

Because of the perishability characteristic of food in a supply chain when moving from upstream to downstream stages and the quantity of food waste, managers of food industries should consider the sustainability of their manufacturing processes and products. Reduction of the amount of waste produced and energy consumed in distribution serve the economic and environmental aspects of sustainability.

Food products decay rapidly during the delivery process; hence, companies should choose strategies that reduce the loss of products and thus their profit caused by the deterioration of food products. Furthermore, it is essential that perishable goods must be delivered within their allowable delivery time windows; otherwise, there may be a penalty for early or late arrivals.

Waste food, air pollution, and greenhouse gas emissions affect human health and contribute to global warming. However, appropriate transportation planning helps decrease them. *Food Supply Chain* (FSC) network design enables an organization to maximize its long-term economic profit and performance. Therefore, sustainability is an essential problem in the perishable food SCM design phase.

Perishable food involves two different concepts of deterioration:

1. Items with a fixed shelf life such as blood, ready-mix concrete, several foods like milk, yogurt with a recommended expiry date.

2. Items with continuous decay such as food, vegetables, flowers, and live animals. In general, quality degradation of food products depends on storage time, storage temperature, humidity, and various factors.

Keeping fresh quality during the transportation of foods is a challenge for food distributors. This issue is addressed based on the assumption that the planning horizon is shorter than the shelf life of the products and the minimization of transportation time and/or distance.

The food distribution is somehow different from the distribution of other products since food products show a gradual quality degradation throughout the supply chain, all the way until final consumption. Hence, in food distribution, quality and safety requirements are more severe. The main difference of food distribution from other products is the continuous process of degradation of food quality; and the dependence of quality and safety on changes in the food product.

Food distribution management is a challenging area that has begun to receive much more attention in the operations management literature than before, due to the following reasons:

- Limited shelf lives of food products
- Environmental requirements, especially about temperature and humidity
- Possible interaction effects among products
- Allowable delivery time for delivering the products
- High customer expectations about the quality (freshness) of products delivered
- Increasing business volume due to the increasing number of end-users as the delivery points

- Low-profit margins

In general, all over the world, food products that are not ending up their life cycles by being consumed have had a significant environmental impact without adding value. This is partly due to food products deteriorating through the supply chain and having to be thrown away. Therefore, sustainability is a prominent issue in the perishable food supply chains.

Decision-making in distribution management is commonly carried out at different decision levels, mainly relating to the time horizon for these decisions. This usually leads to the distinction between long-term, mid-term and short-term planning, or between strategic, tactical, and operational planning. The aim of our study is transportation planning that is concerned with short-term planning of distribution of fresh food and mostly dealing with the planning of deliveries to several customers and routing of vehicles as well as picking up the waste or packaging material back. Typical decisions at this decision level are the details of the delivery routes starting at the depots: at what exact times, by which vehicle, and in which sequence customers will get their products delivered within their allowable delivery times (time windows) without any food waste, and also how much to pick up simultaneously at the customer sites.

In this study, we model the quality degradation as a (continuous) decrease in the product's value (selling price) during its trip that often starts at its highest quality level; however, this might not be the kind of quality decay observed in all food products. A product would be considered wholly perished at a certain quality level long before its shelf life is over because initial quality status may not be easily detectable. It can be hard to estimate the remaining shelf life in such cases.

As one extension of the basic *Vehicle Routing Problem* (VRP), we study the vehicle routing problem with simultaneous pickup and delivery and soft time window (VRPSPD-STW) for perishable food supply chains (VRPSPD-STW-P). We provide a brief background on sustainable supply chains, food supply chains, vehicle routing problem for perishable food, vehicle routing problem with pickup and delivery, and also simultaneous pickup and delivery.

The remainder of this thesis is organized as follows:

In Chapter 2, the literature review is presented. We focus on different distribution and vehicle routing problems with perishable products. We give a review of studies related to the vehicle routing problem and its extensions, including vehicle routing problem with time windows (VRPTW), vehicle routing problem with simultaneous pickup and delivery (VRPSPD), and vehicle routing problem for perishable products.

The problem definition and mathematical formulation of VRPSPD, including soft time windows for perishable products (VRPSPD-STW-P), are proposed in Chapter 3. We present a mathematical model to obtain optimal or near-optimal vehicle routes for a supplier to deliver food products to demand points, mostly retailers.

In Chapter 4, we propose the exact solution approach and heuristic algorithms to solve the VRPSPD-STW-P. Since this problem is NP-hard in nature, algorithms based on various metaheuristics have been widely used in solving the simpler variants of VRP. Hence, we present heuristic and metaheuristic solution methods for solving our problem and improving solutions. We describe three heuristic methods and a genetic algorithm for finding the minimum cost solution for VRPSPD-STW-P.

Chapter 5 presents the experimental studies with the proposed methods. The results of heuristics are compared with the results obtained from the exact solution method. Finally, concluding remarks are given in Chapter 6 that resumes the main findings of our study and point further research issues.

# CHAPTER 2

# LITERATURE REVIEW

The term supply chain management has been defined by Mentzer et al. (2001) as “the systemic, strategic coordination of the traditional business functions and the tactics across these business functions within a particular company and across businesses within the supply chain, for the purposes of improving the long-term performance of the individual companies and the supply chain as a whole”.

We define sustainable supply chain management as, the strategic, transparent integration and achievement of an organization’s social, environmental, and economic goals in the systemic coordination of key inter-organizational business processes to improve the long-term economic performance of the individual and its supply chain. This shows the importance of environmental performance in supply chain partners and that sustainability of any organization is impossible without considering the environmental aspects in the supply chain.

Sustainable supply chain management (SSCM) is the integration of environmental, social, and economic aspects, known as the triple-bottom-line (TBL) dimensions as defined by Elkington (1998). Impacts of supply chain operations on both environment and society force companies to consider TBL effects when designing a sustainable supply chain. Pagell and Wu (2009) deal with strategic trade-offs between the three dimensions of TBL (Figure 2.1).

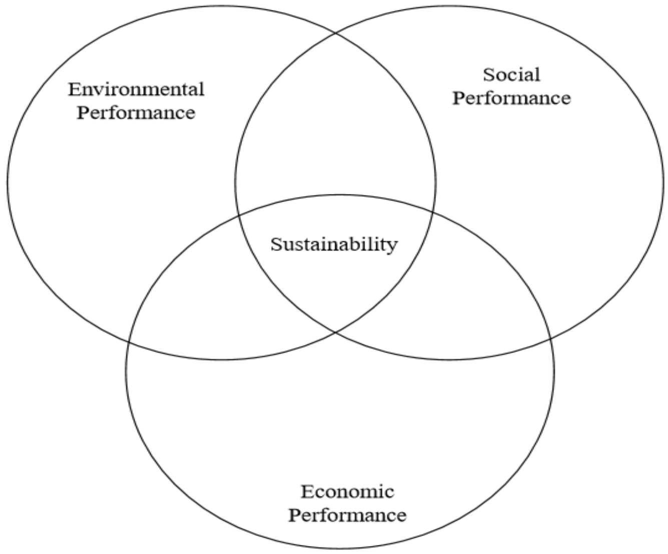



Figure 2.1. Sustainable supply chain management dimensions (TBL)

Seuring and Muller (2008) review 191 papers about the economic and environmental dimensions of the supply chain. Carter and Rogers (2008) provide a comprehensive review of the conceptual aspects of SSCM. Greening supply chains is an important strategy to manage materials flow along value chains to reduce global warming, carbon footprint (Elhedhli and Merrick, 2012). Other environmental issues include waste reduction, transportation costs, reverse logistics, and remanufacturing. A comprehensive discussion for sustainability practices in various industries is presented by Gold et al. (2010) and Wittstruck and Teuteberg (2012).

The goal of a sustainable supply chain is to minimize impacts on the environment while improving environmental performance and maximizing its long-term economic profitability or value (Linton et al., 2007). Hassini et al. (2012) analyze the literature from different perspectives and then provide frameworks and relevant functions for sustainable supply chain management and performance measures. They also provide a case study to illustrate the experience of a utility supply chain in setting performance indicators. Ageron et al. (2012) prepare a systematic review and study the quantitative models of SSCM in several industries. They present a model for sustainable supply management, as seen in (Figure 2.2).

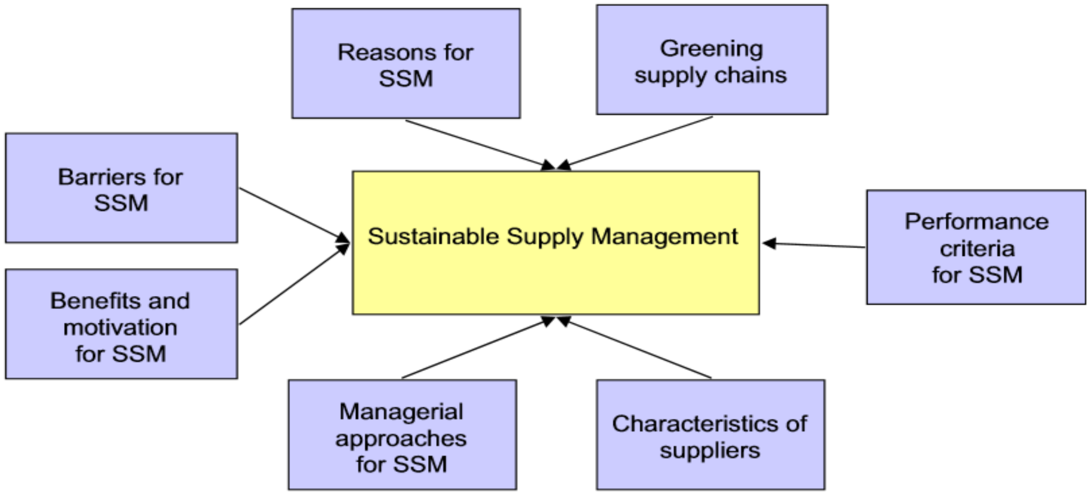



Figure 2.2. A model for sustainable supply management

Fast transportation and low energy consumption are parts of green logistics and SSCM. Srivastava (2007) presents a review that emphasizes green and reverse logistics. Gopalakrishnan et al. (2012) present ten essentialities for deploying sustainability in supply chains (Figure 2.3).

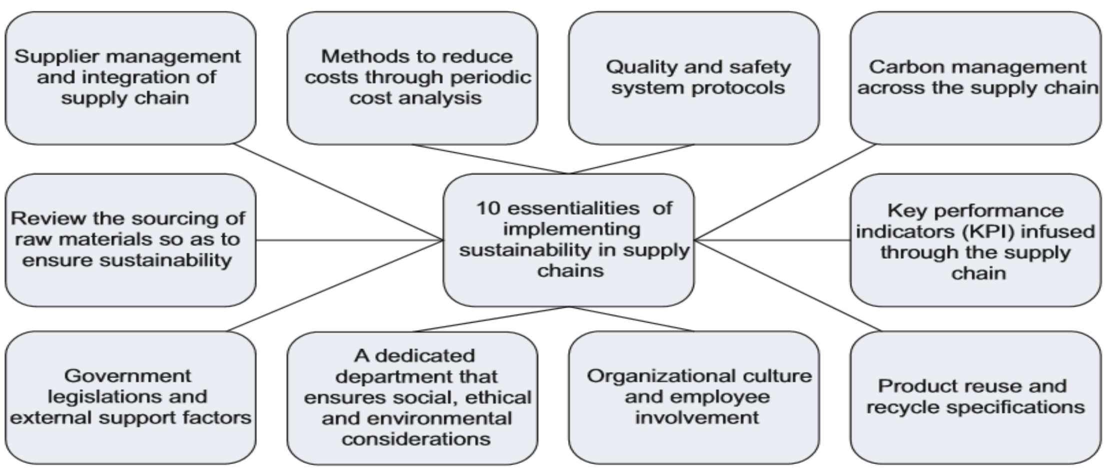



Figure 2.3. Essentialities of sustainable supply chain

Chaabane et al. (2012) propose a Life Cycle Assessment (LCA) methodology for reducing the negative environmental impact of the supply chains. LCA is a process for evaluating the environmental impacts of a product, process, or activity, and an analytical tool that helps assess a product's environmental impact from product development to consumption. It aggregates the results of different aspects of environmental studies, including greenhouse gases (GHG) emissions that are recognized as the most harmful elements to the environment and responsible for climate change. Environmental conditions of storage and transportation facilities have vital importance for the performance of the food supply chain.

Food processing and distribution need different types of chains that are characterized by shorter product lives. This shows the importance of transportation and logistics in the food supply chains (Van der Vorst et al., 2000, Van der Vorst et al., 2009, and Mohan et al., 2013).

Consideration of perishability in the supply chains has received increased attention in both practice and academia (Ahumada and Villalobos, 2009). Food products and production processes have several specific characteristics that influence product quality and quality assurance in the production process (Ziggers and Trienekens, 1997, Van der Vorst et al., 1999, Van Donk et al., 2008, Rong et al., 2011, and Manzizi and Accorsi, 2013). The main target of proposed integrated approaches for supply chain design and management is the simultaneous control of quality, safety, sustainability, and logistics efficiency of food products and processes along the whole food supply chain.

Akkerman et al. (2010) present a comprehensive review of quantitative models on food distribution management. Their focus in food distribution is on food quality, food safety, and sustainability at three decision levels: strategic network design, tactical network planning, and operational transportation planning.

Quality and safety standards in food industries are developed in the last decade (Vellema and Boselie, 2003, and Trienekens and Zuurbier, 2008). Due to the importance of product quality in the food industry, quality assurance in the food industry has become a reality. Based on the requirements of the public sector, private safety and quality standards are being implemented. Trienekens and Zuurbier (2008) discuss public and private standards for the production and distribution of food.

The first comprehensive review for perishable products is given in Nahmias (1982). He reviews the relevant literature on determining suitable ordering policies for both fixed life perishable inventory and inventory subject to continuous exponential decay. Also a brief review of the application of these models to blood bank management is included. An updated review of planning models in the agri-food supply chain can be found in Ahumada and Villalobos (2009).

Goyal and Giri (2001) propose an excellent review of the classification of perishable products and the necessary policies for optimizing the chain. Their study presents a review of the advances of deteriorating inventory literature since the early 1990s. They provide an up-to-date review of deteriorating inventory literature after Raafat's survey (1991). They classify deterioration into ten categories; two main categories of which are as follows:

1. Inventory with fixed lifetime / Inventory with random lifetime

2. Stochastic demand / Deterministic demand

Different transportation and distribution models can be implemented in food supply chains; however, the most effort that has been presented by researchers deals with the critical aspects of the food supply chain. These aspects include procurement, inventory management, order processing, transportation and distribution, sales, demand management, and customer service.

A hierarchical approach is presented by Akkerman et al. (2010). They distinguish three distinct planning levels in distribution management:

1. Distribution network design, concerned with long-term decisions on the physical distribution structure. This includes warehouses and the fleet of vehicles, as well as the related transportation links.

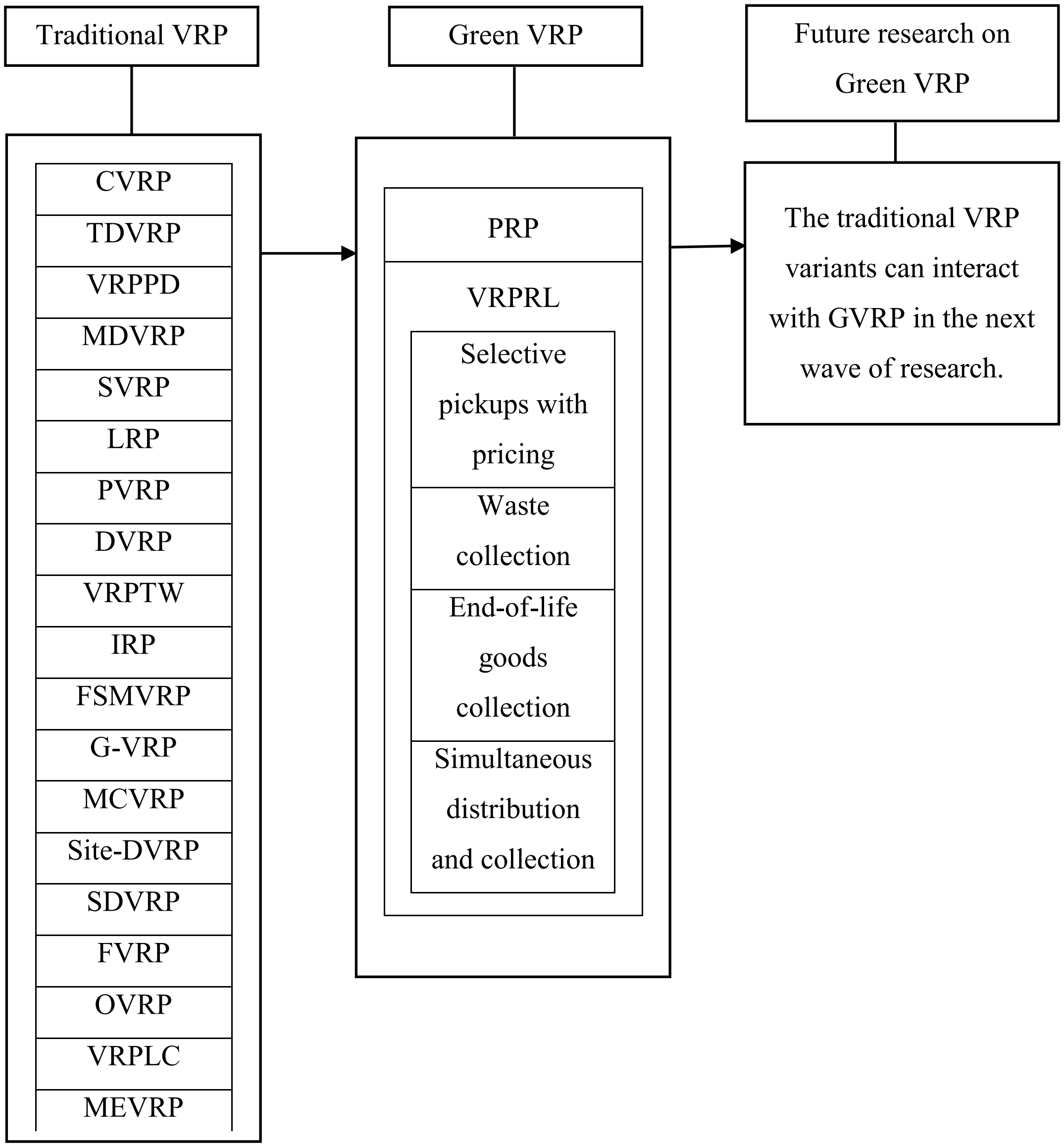
2. Distribution network planning, concerned with mid-term distribution planning decisions related to fulfilling demand at an aggregate level. This includes mainly aggregate product flows and delivery frequencies.

3. Transportation planning, concerned with short-term planning of the distribution of actual customer orders. This includes mainly loading/unloading and routing of vehicles.

There are two studies worth mentioning here for modeling routing for perishable products. Van der Vorst et al. (2009) present a simulation tool for the design of food chains, and Vlajic et al. (2012) define the concept of robustness and study principles and strategies to achieve a robust food supply chain. Robustness is considered as an essential property of supply chains or as a strategy that can be used to improve supply chain resilience. Robustness is related mainly to supply chain vulnerability and uncertainty in general.

To reduce losses of quality, Montanari (2008), Wognum et al. (2011), and Donselaar and Broekmeulen (2012) present advanced inventory and distribution management systems. Donselaar and Broekmeulen (2012) derive approximations for the expected amount of waste in an inventory system with perishable products; thus, they enable retailers to make trade-offs between the relative outdating and the customer service level when making strategic or tactical decisions on the redesign of the perishable inventory system.

Most approaches in short-term planning distribution such as food distribution are based on the well-known vehicle routing problem (VRP), often including delivery time windows. The objective of VRP is to find the minimum distance traveled and/or the minimum number of vehicles. In VRP, vehicle routes should start and end at the same depot; each customer should be visited only once, and the total demand of customers should not exceed the capacity of vehicles. Lin et al. (2014) provide a more recent review for the VRP variants, and the philosophy of their review work is in Figure 2.4.




Note. *CVRP, capacitated VRP; TDVRP, Time dependent VRP; VRPPD, VRP with pickup and delivery; MDVRP, Multi depot VRP; SVRP, Stochastic VRP; LRP, Location Routing Problem; PVRP, periodic VRP; DVRP, Dynamic VRP; VRPTW, VRP with time window; IRP, Inventory Routing Problem; FSMVRP, Fleet Size and Mix Vehicle Routing Problem; G-VRP, Generalized VRP; MCVRP, Multi-compartment VRP; Site-DVRP, Site dependent VRP; SDVRP, Split-delivery VRP; FVRP, Fuzzy VRP; OVRP, Open VRP; VRPLC, VRP with Loading Constraints; MEVRP, Multi-echelon VRP; GVRP, Green VRP; PRP, Pollution Routing Problem; VRPRL, VRP in Reverse Logistics.*

Figure 2.4. The variants of VRP

Some nodes in the distribution networks need to be visited within a particular time range called *time windows* (TW). Solomon (1987) adds time-window constraints to the classical VRP. Two types of time windows are extensively studied in the literature:

1. Hard Time Windows (HTW), where a vehicle must arrive and start service at the customer site within the specified time interval.

2. Soft Time Window (STW), where the time window violation constraint is acceptable but with some penalty.

In the case of soft time-windows, constraints can be violated, but with a penalty cost. When a vehicle arrives with an acceptable delay, the food can still be delivered with a penalty cost. Assume $[a_i, b_i]$ represents the time window of customer $i$; if the vehicle arrives at the customer site before time $a_i$, then it must wait up to time $a_i$ before starting its service. On the other hand, if it arrives after $b_i$, a lateness penalty is incurred; in this case, a lateness penalty is considered because of the customer's loss and short remaining shelf life.

Therefore, if perishable goods are not delivered within the allowable delivery time or time windows in the soft case, penalty costs result for late arrivals. The time window is considered since some part of the revenue is lost due to late delivery. The probability that perishable food can be sold depends on the time that remains between purchasing time and expiration date. This probability decreases at an increasing rate as the time of purchasing becomes closer to the expiration date. Thus, revenue from customers reduces due to late deliveries.

As mentioned before, the increasing environmental and social aspects are forcing companies to consider sustainability in the supply chain. The reverse product flow is also an important relevant contribution to the sustainability of distribution systems. The environmental impact of distribution continues even after a product is delivered to the customer. There is often a reverse flow, ranging from empty recycle containers or boxes in the retail industry to bowls and plates, perished food in the food service industry.

Thus, waste management is the process of protecting the environment and conserving resources. Today's operations research methods and related mathematical models are effectively applied in both forward and reverse logistics. VRP with pickup and delivery (VRPPD) is an extension of the classical VRP. VRPPD is classified into different types; hence we have a brief explanation of them and their underlying assumptions.

In VRPPD, the set of customers is divided into three subsets: the first subset is the linehaul customers with delivery demand only, each requiring delivery of goods from the depot, whereas the second subset of customers is the backhaul customers with pickup demand only, each requiring a pickup of decayed or recycle goods. In the third subset, customers have both linehaul and backhaul demands. In this subset, customers want to receive both delivery and pickup services.

Pickup and delivery problems (VRPPD) can be categorized in four problem types. In the first two problem types, customers have either linehaul demands (delivery) or backhaul demands (pickup) but not both, while in the last two, each customer can have both delivery and pickup demands:

1. *Vehicle routing problem with clustered Backhauls* (VRPCB): All linehaul demands must be satisfied before backhaul demands. The primary assumption of the VRPCB is that all deliveries must be made before any pickups.

2. *Vehicle routing problem with mixed linehauls and backhauls* (VRPMB): VRP with mixed backhauls is the problem in which deliveries and pickups may occur in any sequence on a vehicle route.

3. *Vehicle routing problem with simultaneous pickup and delivery* (VRPSPD): Customers can receive both delivery and pickup services simultaneously in the same vehicle, then each customer must be visited exactly once.

4. *Vehicle routing problem with divisible (split) pickup and delivery* (VRPDPD): Customers demanding delivery and pickup service can be visited more than once; the

class is named as divisible pickup and delivery problems. It is not allowed for the vehicle to simultaneously deliver and pick up goods at the same customer location. If a customer needs both a delivery and pickup service, this customer is to be visited twice in the VRPDPD case resulting in inconvenience and higher transportation costs.

In the next sections, we present a literature review on the main parts of our research. We have a brief overview of the vehicle routing problem (VRP), vehicle routing problem with time window (VRPTW), vehicle routing problem with simultaneous pickup and delivery (VRPSPD), vehicle routing problem with simultaneous pickup and delivery and time window (VRPSPD-TW), and vehicle routing problem for perishable products (VRP-P).

## 2.1. Vehicle routing problem (VRP)

The mTSP (Multi traveling salesman problem) or TSP problem with *m* salesmen reduces to the TSP when there is a single salesman (Bektas, 2006). The mTSP is generally treated as a relaxed VRP where there is no vehicle capacity (Matai et al., 2010). Hence, the formulations and solution methods for the mTSP are also equally valid for the VRP if a capacity is assigned to the salesmen or vehicles.

The paper by Dantzig et al. (1954), the first record in the VRP literature, studies a relatively large-scale TSP and proposes an approximate solution method. Several other TSP articles follow that study. The objective of VRP is to find the minimum distance traveled and/or the number of vehicles. VRP is first introduced by Dantzig and Ramser (1959).

Clarke and Wright (1964) first incorporate more than one vehicle in the problem formulation. Consequently, this study is the first in VRP literature. Other versions of VRP emerge in the early 1970, e.g., routing of public service vehicles (Stricker, 1970), transportation network design (O'Connor and De Wald, 1970), fleet routing (Levin, 1971).

One of the surveys of solution techniques for the VRP can be found in Mole and Jameson (1976). They originally propose the insertion method for solving a traditional VRP. This method incorporates concepts of the nearest neighbor and the sweeping algorithms.

Gavish and Graves (1978) present new formulations for the traveling salesman problem and investigate their relationship to previous formulations. The new formulations are extended to include a variety of transportation scheduling problems, such as the multi-traveling salesman problem and the delivery problem.

Exact algorithms for solving the VRP, based on spanning tree and shortest path relaxations, are discussed in Christofides et al. (1981). The problem of routing a fixed number of trucks, each with a limited capacity from a central warehouse or depot to customers with known demand, is addressed in Golden et al. (1984). They relax the homogeneous fleet assumption. The objective is to determine optimal fleet size by minimizing a total cost function including fixed cost and variable cost components.

VRP related to goods delivery has been extensively examined in Daganzo (1987a, 1987b). Daganzo (1987a) shows how distribution problems with delivery time constraints can be modeled approximately with just a few variables. He suggests that more attention should be paid to the clustering part of algorithm construction. Daganzo (1987b) extends the results of the previous studies concerning the distribution with time windows. This paper discusses different ways of routing vehicles for problems where a significant number of customers do not have time windows. It is shown that a strategy in which all vehicle routes are similar, but customers with time windows and without time windows are differently treated, yields the lowest local distance traveled per point.

VRP research for solution methods accelerates during the 1990s. Primarily due to microcomputer capability and availability, researchers could develop and implement more sophisticated search algorithms. During this era, the term *metaheuristics* is introduced to name several search algorithms for solving these VRPs as well as other combinatorial optimization problems. Laporte and Nobert (1987), and Laporte (1992) survey the main exact and approximate algorithms developed for the VRP. Approximate methods based on simulated annealing and tabu search heuristic for the vehicle routing problem are proposed by Osman (1993). Gendreau et al. (1994), Renaud et al. (1996), Cordeau et al. (1997) describe tabu search heuristics for the VRP. For a further inquiry into major problem types, formulations, and solution methods of this era, one may refer to the works of Laporte and Nobert (1987) and Laporte (1992).

Fisher (1995) presents some variants of vehicle routing problems and various practical issues that arise in the use of vehicle routing models and the most important algorithms that have been developed for the VRP. Golden et al. (1998) present the impact of metaheuristics on solving the vehicle routing problem and algorithms. They investigate the vehicle routing problem and its latest advances.

Bullnheimer et al. (1999) use the ant system to solve the VRP in its basic form. Laporte et al. (2000) study classical and modern heuristics for the vehicle routing problem. The survey is divided into two parts: classical and modern heuristics. The first part contains well-known schemes such as the savings method, the sweep algorithm, and various two-phase approaches. The second part is devoted to tabu search heuristics, which have proved to be the most successful metaheuristic approach.

Toth and Vigo (2003) propose a tabu search algorithm for the VRP. A generalized clustering method based on a genetic algorithm (GA) is developed by Thangiah and Salhi (2001). Baker and Ayechew (2003) describe a genetic algorithm to solve the basic VRP. Prins (2004) presents a genetic algorithm for the VRP, and his proposed algorithm can compete with powerful tabu search algorithms in terms of average solution cost. Prins (2004) algorithm performs better than most published tabu search heuristics on the classical instances of Christofides et al. (1981).

Bell and McMullen (2004) and Reimann et al. (2004) apply ant colony optimization (ACO) to solve VRP. Chu and Prins (2005) develop two insertion heuristics, and a two-phase heuristic called the lower bound heuristic for solving VRP.

Cordeau et al. (2005) review some of the best metaheuristics proposed in recent years for the vehicle routing problem. Eskioglu et al. (2009) propose a taxonomy of the VRP literature, and they have made an effort to classify the literature reviews of the VRP. A review on dynamic VRP can be found in Pillac et al. (2013). The dynamic vehicle routing problem (DVRP) is one of the most important problems in the area of enterprise logistics. DVRP involves these dynamics: the appearance of customers, travel times, service times, or vehicle availability. One of the most often considered aspects of the DVRP is the availability of customers, in which a part or all of the customers are revealed dynamically during the design of the routes. Toth and Vigo (2014) and Lin et al. (2014) provide a new reference study and literature review for vehicle routing problem variants, methods, and applications.

## 2.2. Vehicle routing problem with time windows (VRPTW)

Companies should make both strategic and operational decisions to optimize and manage their logistics system processes efficiently. Studies now focus on more variants of VRP in industries and try to consider different conditions depending on their objectives. Heuristic approaches for VRP do not consider service time intervals or due dates as constraints of the model until Russell (1977), who presents an effective heuristic for the *M*-tour traveling salesman problem and accommodates the time window restrictions in his model.

During the 1980s, surveys generate different static configurations of the original problem of VRP. Some researchers consider vehicle routing problems with time window constraints and construct penalty costs to reflect a violation of these time windows. Schrage (1981) proposes vehicle routing and scheduling problem with time window constraints as an essential area for progress in handling realistic complications and generalizations of the basic routing model. Solomon (1987) adds time window constraints to the classical VRP. He considers the design and analysis of algorithms for vehicle routing and scheduling problems with time window constraints.

Since the VRPTW is an NP-hard problem (Savelsbergh, 1985), exact approaches to solving this type of problem are inefficient in general (Desrochers et al. 1992). The VRPTW has been the subject of intensive research efforts for both exact optimization and heuristic approaches. Early surveys of solution techniques for the VRPTW can be found in Golden and Assad (1986), Desrochers, Kolen et al. (1987), and Soumis (1988). Kolen et al. (1987) use the branch and bound technique to solve VRPTW.

Over the last few years, many authors propose traditional heuristic approaches, for route construction and route improvement (local search). Solomon (1987) develops a few heuristics for solving VRPTW, including savings method, nearest neighbor, insertion, and sweeping. These heuristic algorithms are generally modified from their original versions for solving the traditional VRP. Thus, when applying these algorithms to VRPTW, the feasibility of inserting a retailer node into the route must be checked against the time window constraints. Among the heuristic algorithms developed by Solomon (1987), the insertion method consistently gives excellent results.

One of the methods described by Solomon (1987) is an extension to the savings heuristic of Clarke and Wright (1964). The savings method, which is initially developed for the classical VRP, is probably the best-known route construction heuristic. It begins with a solution in which every customer is supplied individually by a separate route.

Solomon and Desrosiers (1988) survey the significant advances made for the routing problems with time windows, highlighting the significant breakthroughs in solution methodology and their analysis. Koskosidis et al. (1992) focus on determining optimal routes by minimizing total routing costs, including total distance and time costs and cost of waiting due to a vehicle’s early arrival in soft time windows. They heuristically decompose the problem into a clustering component and a series of routing and scheduling components.

Many researchers have made a great effort in solving the VRPTW in recent years by widely used methods based on various metaheuristics. Desrosiers et al. (1995) provide an extensive overview of the models and algorithms for VRPTW, focusing on optimization methods. They show that heuristics remain a viable tool for very large-scale complex problems. Potvin et al. (1996a) and Taillard et al. (1997) use a tabu search for solving VRPTW.

In recent years, metaheuristics like the genetic algorithm (GA) have been widely applied for solving hard combinatorial problems. A genetic algorithm is first proposed by Holland (1975). Since then, GA has been popular due to its contribution to obtaining good solutions for complicated optimization problems in a reasonable amount of time.

In particular, heuristic search strategies based on GA are explored in recent years to improve the solutions in VRPTW. Potvin et al. (1996b), Thangiah (1993), and Thangiah and Petrovic (1998) develop genetic algorithms for the application of new search techniques.

A cutting-plane algorithm is studied in Cook and Rich (1999), and a column-generation-based VRPTW algorithm is presented by Larsen (1999). Cordeau et al. (2001a) present a unified tabu search heuristic for the vehicle routing problem with time windows and for two important generalizations: the periodic and the multi-depot vehicle routing problems with time windows. The significant benefits of the approach are its speed, simplicity, and flexibility. Tan et al. (2001), and Berger and Barkaoui (2003) present heuristic search strategies based on GA to solve VRPTW.

Berger and Barkaoui (2004) provide an improved genetic algorithm for the vehicle routing problem with time windows. They present a parallel version of a new hybrid genetic algorithm for the vehicle routing problem with time windows. The route-directed hybrid genetic approach is based upon the simultaneous evolution of two populations of solutions focusing on separate objectives subject to time constraint relaxation. While the first population evolves the individuals to minimize total traveled distance, the second aims at minimizing time constraint violation to generate a feasible solution.

Kallehauge et al. (2005) focus on the VRPTW as one of the important applications of column generation in integer programming. They discuss the VRPTW in terms of its mathematical modeling, its structure, and decomposition alternatives. They present the master problem and the subproblem for the column generation approach.

Extensive literature reviews on VRPTW and the use of metaheuristics for solving VRPTW up to 2005 can be found in Bräysy and Gendreau (2005a, 2005b). Bräysy and Gendreau (2005a, 2005b) study both hard and soft time windows to solve VRPTW. In Bräysy and Gendreau (2005a) both traditional heuristic route construction methods and recent local search algorithms are examined. The basic features of each method are described, and experimental results are presented and analyzed. The metaheuristic methods are described in Bräysy and Gendreau (2005b). In the case of time windows violation, two typical significant costs are considered:

1. Vehicle costs include fixed costs such as vehicle depreciation and variable costs such as fuel consumption for early arrivals.
2. Delay cost is a penalty that occurs when the vehicle arrives at the customer site later than the time window's end.

Lin et al. (2006) apply the simulated annealing (SA) combined with local search for solving the VRPTW. They show SA as a developed approach to escape from the local optimal traps, and also that the use of exchange and insertion local search can find near-optimal solution efficiently.

Alvarenga et al. (2007) formulate VRPTW as a set partitioning problem and use the Dantzig-Wolfe decomposition method to divide the problem into a master problem and a secondary problem. Their work proposes a heuristic approach for the VRPTW using travel distance as the primary objective through an efficient genetic algorithm and a set partitioning formulation.

Kallehauge (2008) reviews the exact algorithms proposed in the last decades for the solution of VRPTW. They give a detailed analysis of the formulations of the VRPTW and a review of the literature related to the different formulations. He concludes with two main lines of development in relation to the exact algorithms for the VRPTW. One is concerned with the general decomposition approach and the solution to some dual problems associated with the VRPTW. Another more recent direction is concerned with the analysis of the polyhedral structure of the VRPTW.

The hard time window constraint seems to be entirely appropriate to describe the real-world situation. However, sometimes no feasible or executable solution can be obtained if all time window constraints need to be satisfied. Relaxing this strict restriction might result in a better solution concerning the total distance or the total number of vehicles. Furthermore, Montanari (2008) and Tang et al. (2009) show that a tiny deviation from the customer specified time window is acceptable in real life. Adopting soft time window constraints dealing with this possible tiny violation receives close attention in many practical scenarios.

Jiang et al. (2009) propose a particle swarm optimization (PSO) algorithm with crossover for VRPTW. They conclude the PSO algorithm combined with the crossover operation of GA can avoid being trapped in local optimum.

Cheng and Wang (2009) consider different conditions for the VRPTW problem, and they define waiting and delay costs for deviations from the time window in both directions. They investigate time window constraints for VRP in which constraints belong to the soft time windows, and they present GA to solve the problem.

Figliozzi (2009) studies how time window constraints and customer demand levels influence the average distance of VRP. The paper reflects how time window constraints and customer demand levels influence the average distance in VRP, which is an important indicator associated with the decisions in network design, facility location, and fleet sizing, especially for delivering sensitive products. Instead of using traditional heuristics, the study develops a probabilistic modeling approach to approximate the average length of the routes traveled. Figliozzi (2010) proposes to relax hard time windows, leading to lower costs without significantly hurting customer satisfaction.

Qureshi et al. (2009) and Qureshi et al. (2010) study soft time windows where early arrival is allowed at no cost while late arrival incurs a penalty cost. De Oliveira and Vasconcelos (2010) implement a hybrid search method by using an efficient simulated annealing algorithm. Zhen (2011) studies the multi-period vehicle routing problem with recurring dynamic time windows as an extension of VRPTW and proposes a time window partitioning method for the problem. Qureshi et al. (2012) propose a model for dynamic VRP with soft time windows to help freight carriers avoid extra cost as well as lateness of goods delivery.

### 2.3. Vehicle routing problem with simultaneous pickup and delivery (VRPSPD)

In the VRP literature, there is a mass of studies in VRP with pickup and delivery, such as VRP with simultaneous pickup and delivery. Many applications of the classical VRP involve pickup and delivery services between the depot and some peripheral locations as warehouses, stores, and stations. The extension models of VRP, such as VRPCB and VRPMB, release the capacity and number-of-visits constraints of the traditional VRP that customers' total demands cannot exceed a known quantity due to vehicle capacity, and each customer should be visited only once. The mathematical formulation for VRPSPD is first introduced by Min (1989). The work is concerned with a library situation where delivery and pickup of books are required.

Mosheiov (1998) studies a different version of the vehicle routing problem with pickup and delivery that was not surveyed before. Delivery, in his case, refers to the transportation of goods from the depot to customers, and pickup refers to shipment from customers to the depot. He develops an alternative solution method, which is an extension of the well-known tour-partitioning heuristic. Tour partitioning heuristics for solving the capacitated vehicle routing problem are based on breaking a basic tour into disjoint segments served by different vehicles.

Salhi and Nagy (1999) propose four insertion-based heuristics for generating solutions to VRPSPD. Dethloff (2001) proposes a mathematical formulation for VRPSPD and an insertion-based heuristic algorithm to solve the problem. Dethloff (2002) compares the insertion-based algorithm with the other algorithms that are originally developed to solve VRPSPD.

Nagy and Salhi (2005) also propose a local search heuristic with four phases to solve VRPSPD. They show that the VRPSPD is a generalization of the mixed linehauls and backhauls problem. They also extend their method for the multi depot case.

Dell'Amico et al. (2006) present an optimization algorithm based on column generation, dynamic programming, and branch and price method for solving VRPSPD. Montané and Galvão (2006) present an extension of the formulation by Mosheiov (1994), and Gavish and Graves (1978) formulations. For obtaining lower bounds for these problems, Montané and Galvão (2006) develop a second formulation that includes maximum distance constraint, and they present a tabu algorithm to solve VRPSPD. Their algorithm uses three types of movements to obtain inter-route adjacent solutions: the relocation, interchange, and crossover movements.

Parragh et al. (2008a) and Parragh et al. (2008b) present two comprehensive surveys and classification schemes for pickup and delivery problems. Parragh et al. (2008a) refer to all those problems where goods are transported between pickup and delivery locations. Parragh et al. (2008b) discuss the transportation of goods from the depot to linehaul customers and from backhaul customers to the depot. Chen and Wu (2006) propose a heuristic method based on tabu search and route improvement procedures for VRPSPD. Cao and Lai (2007a) use an improved evolution algorithm to solve VRPSPD. An improved genetic algorithm is proposed based on the crossover operator to solve the problem.

Alshamrani et al. (2007) examine a real-world problem of blood distribution and collection of blood containers. They generate a penalty cost if the containers are not picked up. Bianchessi and Righini (2007) use Dethloff (2001) VRPSPD formulation and develop heuristic algorithms for solving VRPSPD. Their work comprises of different constructive algorithms, local search algorithms with various neighborhood structures, and tabu search algorithms.

Ganesh and Narendran (2008) develop a mathematical programming model and a two phase heuristic to solve the VRPSDP. In the first phase, they use a heuristic to find an initial solution. In the second phase, they develop an enhanced version of simulated annealing (SA) to search for the best solution. Wassan et al. (2008) develop a tabu search algorithm to solve VRPSPD.

Ai and Kachitvichyanukul (2009) reformulate the VRPSPD as a direct extension of the basic VRP. As a result of their survey, the formulation of Min (1989), Dethloff (2001), and Montané and Galvão (2006) can be reduced to a special case of their reformulation. They present a solution method based on the particle swarm optimization algorithm for VRPSPD.

Many researchers have put their efforts into seeking a solution to VRPSPD in recent years, and they use metaheuristics to solve VRPSPD. Gajpal and Abad (2009) solve the VRPSPD through an ant colony algorithm with a construction rule and two multi-route local search schemes. Çatay (2010) proposes an updated procedure for the VRPSPD, ant colony algorithm with a saving-based function.

Subramanian et al. (2010) present a parallel approach for solving VRPSPD. The parallel algorithm is embedded with a multi-start heuristic, which consists of a variable neighborhood descent procedure, with a random neighborhood ordering integrated into an iterated local search. Subramanian et al. (2011) propose a branch-and-cut with a lazy separation approach over the extended flow formulations that use the separation routines for the capacitated vehicle routing problem. They test the algorithm in a large-scale logistics network, and the results demonstrate that their approach can improve most of the previously known bounds.

Zachariadis and Kiranoudis (2011) suggest a local search based metaheuristic. Their proposed VRPSPD heuristic is a local search algorithm which makes use of two algorithmic concepts, exploring solution neighborhoods, and avoiding search cycling. Tasan and Gen (2012) develop a GA for VRPSPD. In their study, the length of the chromosome is determined by the number of customer nodes which are served by the vehicles; routes are determined based on the capacities of vehicles. In the proposed methodology, initial population generation process is based on random permutation. Gan et al. (2012) apply multi cooperative particle swarm optimization algorithm to solve VRPSPD.

Zhen et al. (2020) define a new variant of the vehicle routing problem by combining the consistent VRP and the VRP with simultaneous delivery and pickup. They highlight the necessity of considering the service consistency in reverse logistics by considering the consistency of drivers and the arrival time, as well as the simultaneity of delivery and pickup. To solve this problem, three heuristics are proposed on the basis of the record-to-record travel algorithm, the variable neighbourhood search, and the tabu search-based method. Koç et al. (2020) present a comprehensive review for the existing work on the VRPSPD. They survey mathematical formulations, algorithms, variants, case studies, and industrial applications. They also provide an overview of trends in the literature and identifies several interesting future research perspectives.

## 2.4. Vehicle routing problem with simultaneous pickup and delivery and time window (VRPSPD-TW)

VRPSPD-TW is an extension of VRPTW, where customers require simultaneous pickup and delivery. The problem is a combination of the standard versions of VRPSPD and VRPTW and is denoted by VRPSPD-TW. The VRPSPD-TW considers simultaneous pickup and delivery at each customer such that a customer is visited only once within the specified time window without violating the vehicle capacity constraints. Research on VRPSPD-TW has been very recent and thus limited.

Sexton and Choi (1986) is one of the first surveys of VRSPD-TW. They consider a problem called the single-vehicle routing and scheduling problem with soft time windows, partial loads, and dwell times (the amounts of time required to load and unload at each origin and destination). They apply Benders' decomposition procedure and construct a route improvement heuristic based on the master problem.

Gelinas et al. (1995) study VRP with backhauls and time windows (VRPBTW). They present a new branching strategy for branch-and-bound approaches based on column generation for the vehicle routing problem with backhauls and time windows. This strategy involves branching on time or capacity variables rather than on network flow variables and also presents criteria for selecting network nodes for branching.

Thangiah et al. (1996) describe a route construction heuristic for the VRPBTW, as well as different local search heuristics to improve the initial solution. They propose a heuristic based on the insertion procedure to obtain initial solutions to the VRPBTW. Then, the initial solutions are improved through the application of two points crossover procedure.

Angelelli and Mansini (2002) provide two mathematical programming models to represent the VRPSPD-TW. They use a column generation method based on set covering formulation. Ropke et al. (2007) describe a branch-and-cut algorithm for the pickup and delivery problem with time windows. They introduce several valid inequalities as cutting planes, and these inequalities are used within branch-and-cut algorithms which are tested on several instance sets.

Cao and Lai (2007b) investigate VRPSPD-TW and propose a heuristic method based on a genetic algorithm to solve this NP-hard problem. Wang (2008) studies a tabu search algorithm, and Mingyong and Erbao (2010) propose a differential evolution algorithm to solve VRPSPD-TW with limited route distances. Differential evolution algorithm combines simple arithmetic operators with the classical operators of crossover, mutation, and selection to evolve from a randomly generated starting population to a final solution. The route length constraints reduce the search space in obtaining optimal or near-optimal solutions.

Gan et al. (2012) consider two additional factors of VRPTW, which are the unknown number of vehicles and simultaneous pickup and delivery service. They apply an efficient multi-swarm cooperative particle swarm optimization algorithm, and propose a new encoding method.

Kassem and Chen (2013) provide a formulation for VRPSPD-TW. They consider a connected network where at each node a customer requires a certain amount of new products to be delivered from a central depot in the network and a certain amount of end-of-life products to be collected from her site and returned to the central depot. They develop a heuristic solution methodology to solve VRPSPD-TW. The proposed solution methodology consists of a sequential route construction algorithm to solve VRPSPD-TW. The algorithm begins by selecting a seed customer to build a route from the depot to that seed customer and back to the depot. Then, according to insertion criteria, another customer is chosen to be inserted into the route. The insertion process continues until either all customers are included in the route and an initial solution is generated, or it is not feasible to insert new customers into the current route. Then a simulated-annealing-based search process is used to improve the initial solutions.

Shi et al. (2020) study the VRPSPDTW with two objectives: a primary objective for minimizing the number of vehicles and a secondary objective for reducing the transportation distance. They propose an effective learning-based two-stage algorithm, in which a modified variable neighborhood search is proposed to minimize the primary objective in the first stage and a bi-structure based tabu search is designed to optimize the primary and secondary objectives further in its second stage.

**2.5. Vehicle routing problem for perishable products**

Perishable goods such as food products and vegetables deteriorate during the distribution process. Suppliers should adopt a well-designed strategy that reduces the loss of their profit due to the deterioration of perishable goods. Delivery planning is quite a critical issue in perishable food supply chains. The mathematical models developed so far for food distribution usually consider the perishable nature of food products.

Tarantilis and Kiranoudis (2001) develop a heterogeneous fixed fleet vehicle routing problem and examine tour improvement procedures for solving the fresh milk distribution problem. These algorithms are also characterized as local search algorithms, which represents the solution of the problem, by performing changes-moves, taking a customer from its position in one route and moving it to another position in the same route or different routes. Tarantilis and Kiranoudis (2002), in a later study, propose a multi-depot vehicle routing problem for the distribution of fresh meat.

Zhang et al. (2003) show that controlling product quality throughout the supply chain requires a focus on both time and temperature. They consider a three-level distribution system with fixed plant locations, and distribution warehouses as well as retailers. In their study, product quality is represented as a function of time and temperature for production, transportation, and storage. They discuss a tabu search procedure to optimize the structure of cold chains by minimizing storage and transportation costs.

Prindezis et al. (2003) present an application service provider that would offer logistics for central food markets by appropriately solving the fleet management problem. This system automatically generates vehicle routing plans such that all customer's demands are met, no constraints are violated, and a combination of vehicle costs and distance traveled is minimized. Smith and Sparks (2004) consider product quality as one of the essential food product characteristics throughout the supply chain, which degrades depending on the environmental conditions of storage and transportation facilities.

It is difficult to define the quality models for perishable foods since food degradation depends on various parameters, and there exist many criteria on how to measure it. Perishable food deteriorates because of bacteria, light, and air; higher temperature results in a higher rate of spoilage. Different transportation models can be implemented in food supply chains. Verbic (2004) and Bogataj et al. (2005) have studies on estimating the quality of the perishable products and maintaining foods in high quality. These methods are very sophisticated and require the estimation of many parameters that are not directly available in the distribution process.

Meat is a highly perishable food product that, unless correctly stored, processed, packaged, and distributed, spoils quickly and becomes hazardous due to microbial growth. Belenguer et al. (2005) present a computer program that has been developed to design the dispatching routes of a medium-sized meat company in Spain. They formulate the real problem as a variant of the vehicle routing problem with time windows.

In Lutke Entrup et al. (2005) and Eksioglu and Jin (2006), and Ahuja et al. (2007), product quality is implicitly considered by constraining the shelf life of the product. As with all fresh products, yogurt has a relatively short shelf life. The manufacturer usually determines the shelf life of a product. The attempt to deliver the yogurt products as freshly as possible to the retailer has significant effects on the supply chain.

Lutke Entrup et al. (2005) present mixed-integer linear programming models that integrate shelf-life issues into production planning and scheduling in yogurt production. Eksioglu and Jin (2006) address a production and distribution planning problem in a dynamic two-stage supply chain with a perishable final product, which has a limited shelf life. They discuss a planning model that integrates production, inventory, and transportation decisions in a two-stage supply chain for perishable products. Ahuja et al. (2007) study a two-stage logistic network similar to Eksioglu and Jin (2006) with additional production and inventory capacity constraints.

Privé et al. (2006) solve a VRPSPD and mixed fleet VRP arising in soft-drink distribution and collection of recyclable containers in a company. They use the nearest-neighbor heuristic to generate initial solutions set and an improvement phase with three steps,  three-point crossover, two interchange procedures, and route merge. Naso et al. (2007) consider the problem of scheduling the production and distribution activities of a network of plants that are supplying rapidly perishing materials. They develop a genetic algorithm for supply chain scheduling, and they study a case study in the distribution of ready-mixed concrete-like cement production.

Van der Vorst et al. (2007) present one of the keys for an integrative view on logistics and product quality for the food industry, labeled 'quality-controlled logistics' in SCM. A vehicle routing problem with time window for perishable food is constructed in Hsu et al. (2007) by considering the randomness of the perishable food delivery process. They assume a decrease in the value for perishable food that has to be stored at chilled temperatures throughout their lifetime and that the rate of deterioration is dependent on the temperature at any moment.

Zanoni and Zavanella (2007) develop a model by considering fixed shelf-life product quality and present heuristic algorithms to minimize storage and transportation costs. In their later study, Zanoni and Zavanella (2012) provide the strategic role of energy in different food supply chains. The study shows how the energy effort produced in cooling and maintaining products plays an essential role in the food supply chain’s effectiveness and sustainability. The model allows understanding the relationships between quality, temperature, and energy, addressing a possible approach to chain optimization.

Osvald and Stirn (2008) develop a heuristic algorithm for distributing fresh vegetables and similar perishable food in which perishability represents a critical factor. The problem is formulated as a VRPTW and with time-dependent travel times. The model considers the impact of perishability as a part of the overall distribution costs. To measure the decrease in the value of a load of fresh vegetables, they define the quality of the load by extending the simple linear model proposed by Pawsey (1995) to estimate the decrease of fresh vegetable quality. The quality is 100% when the load can be sold entirely at the current market price, and the quality is 0% when the load has lost its commercial value.

Chen et al. (2009) propose a specific design for a chain for perishable goods and present a nonlinear model for production scheduling and vehicle routing problem with time windows in which penalty should be incurred for late arrivals. Demands at retailers are assumed stochastic; and the quality of products decreases throughout their lifetimes with a fixed rate of deterioration. They assume soft time windows and the vehicle that arrives late incurs a penalty. The problem is studied in two cases: deterministic vehicle travel time and stochastic vehicle travel time. Traveling costs depend on the distance traveled, and inventory costs depend on a time-dependent deterioration function for perishable food.

Ahumada and Villalobos (2009), Akkerman et al. (2010), and Karaesmen et al. (2011) present comprehensive reviews of quantitative models on food distribution management. Akkerman et al. (2010) focus on three aspects of food distribution: food quality, food safety, and sustainability in three decision levels, including strategic network design, tactical network planning, and operational transportation planning.

It is clear from the literature review that incorporating the perishability factor explicitly in the formulations seems to be of great advantage since the customers' point of view in terms of service is also taken into account. In many countries, around 30% of food products are wasted throughout the supply chain (Chapman, 2010), even though a large part of this waste occurs at the final consumer and retailers. The determination of shelf life as a function of variable environmental conditions has been the focus of many research activities in this field, and a considerable number of reliable models have been developed. These models consider the knowledge about microbial growth in decaying food products under different temperature and humidity conditions.

Ahumada and Villalobos (2011) present an integrated planning model for the production and distribution of fresh products, and they consider the distribution of the crop. The model considers the perishability of the crops in two different ways, as a loss function in its objective function and as a constraint for the storage of products.

Rong et al. (2011) develop linear and exponential product quality degradation models in single-product production and distribution planning models based on mixed-integer linear programming. They design a multi-period modeling approach for presenting the dynamics of the decay. Their distribution model uses the predictive microbiology knowledge in forecasting shelf-life based on the temperature of transportation and stocking.

Yan et al. (2011) consider an integrated production-distribution model for a deteriorating item in a two-echelon supply chain. Their results indicate that when the deterioration rate goes up, the production lot size and the corresponding cycle time are reduced to benefit the entire supply chain. Their objective is to minimize the total system cost while some restrictions relating to perishability are imposed.

Amorim et al. (2011) propose an integrated production and distribution planning model for perishable products. Amorim et al. (2012) propose an integrated production and distribution planning through a multi-objective framework. In the first objective total costs are minimized, namely: production costs, transportation costs, and spoilage costs. In the second objective, the mean fractional remaining shelf life of products to be delivered is maximized. They formulate models for the case where perishable products have a fixed and a loose shelf life (i.e., with and without a best before date). Their results show that the economic benefits derived from using an integrated approach are dependent on the freshness level of products delivered.

Amorim et al. (2013) provide a review that classifies production and distribution planning models dealing with products subject to physical deterioration. Throughout the review, they point out a specific approach that perishability may enforce in such models and how mathematical modeling techniques can address a wide range of different perishability forms.

Amorim and Almada-Lobo (2014) study a multi-objective model that includes a distribution cost minimization function and freshness state of the delivered products as maximization objective function. As soon as the vehicle departs from the depot, all products that are carried are at their maximum freshness. Their formulation and notation are based on the VRPTW formulation proposed in Cordeau et al. (2001b).

# CHAPTER 3

# PROBLEM DEFINITION AND MATHEMATICAL FORMULATION

In this chapter, we define our problem for the distribution of perishable food and picking up the out-of-date (spoilt) food within the time windows of customers, then formulate it as a mixed-integer linear programming model. Specifically, we deal with a type of vehicle routing problem (VRP) with soft time windows to consider the perishable nature of the products delivered. We aim at finding an optimal set of vehicle routes for a particular fleet of vehicles that serve a given set of customers with known delivery and pickup demands so as to minimize total transportation, time window violation, and quality-related costs.

Consider a fleet of identical vehicles, each with capacity $Q$. The classic vehicle routing problem (VRP) is a combinatorial optimization problem that results in proposing a set of routes, each of which includes a set of edges, several vehicles departing from the depot serving a series of customers at the vertices, and returning back to the same depot.

Each customer should be visited only once in the route, and the demand of a customer should not exceed the vehicle capacity. Some formulations for VRP consider a start at depot 0 and end at depot $n + 1$. If each vehicle should come back to the start depot, then $n + 1 \rightarrow 0$. Each customer has a demand that needs to be satisfied for a specific product. The VRP deals with satisfying a set of customers' demands by a set of vehicles departing from the depot. The objective of the problem is to find a set of routes that satisfy all customers' demands and minimize the total distance traveled or traveling cost.

Since the freshness of the food products is essential, retailers are very demanding for the on-time delivery of the food products. Moreover, each retailer has its desired time window for receiving the products according to operational requirements, traffic conditions, or government regulations. As we defined before, such time windows turn out to be additional constraints for VRP. The vehicle routing problem with time windows is an extension of the VRP. The VRP with time window constraints is referred to as the vehicle routing problem with time windows (VRPTW).

The transportation of waste or spoilt products and end-of-life (EOL) products are part of Green Logistics. Many distribution planning problems, such as vehicle routing problems with pickup and delivery, consider these reverse logistics to avoid any damage to the environment through recycling, remanufacturing, reusing products and materials, and even upcycling.

Collecting spoilt and end-of-life products from customers may be in different ways in vehicle routing. One of the applicable versions is the vehicle routing problem with simultaneous pickup and delivery (VRPSPD), in which a customer is visited only once, and deliveries and pickups are simultaneous. Researchers focus more on the applicability of VRPSPD in industries and society by considering time windows, multiple vehicles, and multiple depots since companies should make strategic and operational decisions to optimize and efficiently manage their logistics systems. However, we acknowledge that, for food products, supply chains are organized differently in such a way that they are likely to be shorter because of perishability, so the quality level of products upon arrival at the customer is crucial and should not be overlooked in the design of food supply chains.

The general constraints for a classic VRPSPD are as follows:

- Each route begins and terminates at a central depot.
- Each customer is visited exactly once by one vehicle only.
- Every vehicle in each route transports the delivery demands from the depot to the customers assigned.

- Every vehicle in each route transports the pickup demands from its assigned customers to the central depot.
- The capacity of vehicles should not be exceeded.

We study an extended formulation of the VRPSPD, which has soft time windows for the customers and is related to the distribution of perishable products (VRPSPD-STW-P). We have a brief overview of the VRPSPD-STW-P and its general formulation in the following sections.

## 3.1. The VRPSPD-STW-P

The mathematical formulation for the VRPSPD is first introduced by Min (1989). The goal of the VRPSPD is to find the number of vehicles needed to serve all customers, an optimal set of vehicle routes for the vehicles of the fleet, which need to serve a given set of customers so as to minimize total transportation and vehicle-related costs, penalty costs for late arrivals, and loss-of-quality cost during storage and transportation.

Our VRPSPD-STW-P solution consists of routes such that:

- Each route starts and terminates at a single central depot.
- A route length should not exceed a specified length (either in distance or traveling time).
- A single product or an aggregated product is delivered with a shelf life and therefore deteriorates as time passes by.
- The number of vehicles allowed for use is sufficient to guarantee a feasible solution.
- A vehicle completes only one route from the depot to its assigned customers and back to the depot.
- The vehicles in the fleet are homogeneous.
- Each customer is visited exactly once by one vehicle only for both pickup and delivery.

- Each customer should receive her ordered products within the appointed time window with its preferred quality level.
- Both the demands to be delivered and the quantities to be picked up are known in advance.
- Every vehicle in each route transports the delivery demands from the depot to the customers assigned to it as defined by its route.
- Every vehicle in each route transports the pickup demands from its customers back to the central depot.
- Each customer's pickup quantity should be picked up immediately after its demand is delivered; hence delivery and pickup at a customer are simultaneous.
- The vehicle spends a specific service time, including unloading of the product delivered and picking up the end-of-life product at each customer on its route.
- The capacity of a vehicle should not be violated.

In the remainder of this section, we give a brief background on time windows, linear quality degradation of food products and acceptable quality level for customers.

### 3.1.1. Hard and soft time windows

Every customer has a specified time zone for receiving the product due to her operational requirements, quality concerns, traffic conditions, and governmental rules and regulations. Such time zones are called time windows and form the additional constraints for any VRP.

We study VRPSPD-TW with time windows by considering the characteristics of perishable food delivery. Since the freshness of food products is essential, customers are assumed to be very sensitive for the delivery time of the products. Delivery failure is defined as the condition when the customer does not receive the ordered products within the appointed time window with her preferred quality level.

In practice, the severity of time windows can vary depending on customer requirements. In the extreme case, the vehicle must arrive exactly within the time window, and vehicles that violate this requirement are rejected ('hard' time window constraints). On the other hand, in some cases, violation of the time window is acceptable to some extent, but a penalty is applied for violating the time window ('soft' time window constraints).

We study soft time windows in our problem. In this soft time window case, early arrival is allowed at no cost, while late arrival incurs a penalty cost. Assume $[a_i, b_i]$ represents the time window of customer $i$, and $a_i$ and $b_i$ are the beginning and the end of the specified delivery time imposed by the $i$th customer. If a vehicle arrives at the customer before time $a_i$, it must wait up to time $a_i$ until starting its service, which is allowed at no cost. On the other hand, if it arrives after $b_i$, a lateness penalty cost is incurred for each unit time of lateness.

### 3.1.2. Quality loss for perishable products

It is usually difficult to define the quality models for perishable foods since food degradation depends on various parameters, and there are several measures for quality assessment. The value or quality of perishable food products decreases rapidly once produced and keeps decaying while being delivered. The revenue of the food supplier depends on the condition of the products when the customers receive them. Thus, timely production and delivery of perishable foods significantly affect the supplier's revenue.

We present a study of the parameters affecting the preservation of perishable products in order to keep them at the required level of quality and quantity for the final delivery. Most of the methods are very complex and require the estimation of many parameters, which may not be available in the distribution process.

We assume that any perishable product has a limited shelf life under the defined conditions. We can use an absolute or a logarithmic quality measure so that the resulting quality level degrades linearly over time. It means that the quality degradation is associated with the sum of storage and transportation times. Therefore, we use the simple linear model proposed by Pawsey (1995) to estimate the degradation in the quality of fresh food product.

Perishability concerning time can be formulated as follows:

The point $t$=0 represents the highest quality level (freshest) of the products. In this case, we assume the perishability starts at $t$=0. The first stage is the time from $t$=0 until the departure time from the depot, that is the waiting time in the depot before departure.

The second stage includes the transportation time from the depot to the customer. Time window constraints are said to be soft because the start and end times of the time window are somehow relaxed. The vehicle is allowed to wait in the case of early arrival at the customer node. Then, we have the third stage, which is the quality reduction from the arrival time at the customer node until the start time of the time window, $a_i$.

We define the variables for quality levels of products and denote them by $q$ for the time interval from $t$=0 up to the starting time of service at the customer site to reflect the perishability effects over time. These $q$ variables have a direct relationship with both the shelf life of product as a time parameter and the time at which service is started at the customer.

Thus, by minimizing the loss of quality for the products delivered, it is tried to deliver products at their maximum possible quality. A loss of quality of 20% for example, can be related to either a vehicle load where 20% of transported products is entirely damaged, and 80% of them is in perfect condition, or all product transported were evenly damaged, so they could be sold only at 80% of their original price. In this study, we consider the latter assumption that the whole transported products are evenly damaged so that they could be sold only at $q$ % of their original price.

We estimate the linear dependency of quality on time. The loss of quality represents an additional cost for the sellers (the complete chain from depot to customers in the distribution process) because some portion of the products cannot be sold in the market, or it can be sold but at a lower price due to loss of quality.

The deterioration of quality over time depends on the storage conditions, type of products transported, traveling time, and service start time at the customer. Assuming proper storage conditions during travel time, we consider loss of quality by means of the reduced revenue in the objective function.

The quality level function is assumed to vary between 0% and 100%, where 100% corresponds to the maximum possible quality, at $t$=0. The quality level of the perishable products decreases as time goes on, no matter whether they are kept in storage or delivered to customers.

Each customer $i$ has her own minimum preferred quality level denoted by $ql_i$, and the maximum preferred quality level denoted by $qu_i$, which is actually 100%. $ql_i$ is based on the quality level imposed by customers based on the average of the maximum and minimum times for delivery of the product to customer $i$.

It should also be noted that it is unnecessary to put any limit on the maximum preferred quality level because higher $qu_i$ gives more satisfaction to the customers.

In the next section, we formulate this vehicle routing problem for simultaneous pickup and delivery of perishable products with soft time windows (VRPSPD-STW-P). We first discuss the assumptions underlying this variant of VRP from the basic VRPSPD-TW.

### 3.2. MILP model for VRPSPD-STW-P

We develop a mixed-integer linear programming (MILP) model to formulate the VRPSPD-STW-P. We present our model, VRPSPD-STW-P, with mainly four extensions of the VRPSPD-TW formulation:

1. Relaxation of the hard time window to the soft time window by enlarging the hard time window

2. Restriction on the number of available vehicles in the depot and allowance for rented vehicles at the depot

3. Perishability of the product starts from $t$=0, and goes on during storage time and travel time until starting service at the customer. In the case of early arrival, there is a waiting time until the beginning of the preferred delivery time (beginning of the time window). However, in the case of late arrival, the time at which the vehicle starts servicing the customer and arrival time at the customer is the same. All in all, we assume the perishability of food starts at $t$=0. Then quality degrades in the depot before departure (unless the departure time from the depot is $t$=0) until the start of service at the customer.

4. Range of the quality level

**Sets:**

Let $G$ = ($\acute{N}$, $A$) be a directed graph where $\acute{N} = N \cup \{0\}$ is a vertex set with $N = \{1,2,\dots,n\}$ and $A = \{(i,j): i,j \in \acute{N}\}$ is an arc set. $K$ is a set of identical fixed capacity vehicles with $K$={1,2,…,m} that are initially located at the depot represented by the vertex 0, and all vehicles are available to deliver products through a set of routes (arcs) to a set of customers (vertices).

**Indices:**

$k$: vehicles, $K = \{1,2, \dots, m\}$

$i, j$: vertices (nodes), $\acute{N} = N \cup \{0\}$ and $N = \{1,2, \dots, n\}$, $|\acute{N}| = n + 1$, $i, j \in \acute{N}$

$(i, j)$: arcs

**Parameters:**

$Q$: vehicle capacity

$s_i$: service time at customer $i$

$w_{ij}$: travel distance from customer $i$ to customer $j$

$t_{ij}$: travel time from customer $i$ to customer $j$

$a_i$: start of time window for arriving at customer $i$

$b_i$: end of time window for arriving at customer $i$

$e_i$: earliest acceptable time for early arrival at customer $i$

$l_i$: latest acceptable time for late arrival at customer $i$

$d_i$: demand quantity of customer $i$

$z_i$: pickup quantity of customer $i$

$p$: selling price of one-unit food product at the depot

$c$: transportation cost per unit distance travelled by an owned/rented vehicle

$c_b$: lateness penalty cost per vehicle per unit of time

$sl$: shelf-life of perishable food product

$L$: maximum length of each route

$ql_i$: lowest quality level acceptable by customer $i$

$f_a$: fixed cost for a vehicle available at the depot

$f_r$: fixed cost for a rented vehicle

$m_a$: number of vehicles owned and available at the depot

$m_r$: number of rented vehicles at the depot

$m$: minimal number of vehicles to ensure a feasible solution

$M$: sufficiently large positive number

**Decision Variables:**

$x_{ijk}$: equals 1 if arc $(i, j)$ is traveled by vehicle $k$, 0 otherwise

$T_{ik}$: time when vehicle $k$ starts servicing customer $i$

$y_{ijk}$: load of vehicle $k$ when vehicle $k$ departs from customer $i$ toward customer $j$

$q_i$: final quality level of the product when service starts at customer $i$

### 3.2.1. Objective function

The objective function consists of the following cost components:

- Variable transportation cost
- Fixed vehicle cost for both the owned and rented ones
- Penalty cost for late arrivals at customers
- Loss-of-quality cost during storage and transportation from $t$=0 until starting service at customers.

In the following subsections, we describe each cost element of the objective function.

- **The variable transportation cost**

The total variable transportation cost, $TC_{tr}$, is formulated as

$$TC_{tr} = c \sum_{i \in \acute{N}} \sum_{\substack{j \in \acute{N} \\ i \neq j}} \sum_{k \in K} w_{ij} x_{ijk}$$

- **The fixed vehicle cost**

The number of vehicles required in the model is a parameter that is determined in advance by trial-and-error until the minimal number of vehicles to obtain a feasible solution is found out. The reduction in the number of vehicles may contribute to reducing the total costs, which is quite simple; but it may not meet the requirements of time window, quality, and route length. Thus, we try to obtain the minimum number of vehicles that satisfy all constraints, hence yields a feasible solution.

The variable $m$ is the total number of vehicles required to guarantee a feasible solution: the sum of owned and rented vehicles. The minimum number of vehicles under this condition should satisfy the constraint below as follows:

$$\sum_{j=1}^{n}\sum_{k=1}^{m} x_{0jk} = \sum_{j=1}^{n}\sum_{k=1}^{m} x_{j0k} \leq m$$

This constraint guarantees that the number of vehicles departing from the depot must be less than or equal to the total number of vehicles available (owned and rented) at the depot for attaining a feasible solution.

The number of owned vehicles in the depot is known and denoted by $m_a$. If the number of vehicles required to obtain a feasible solution is more than $m_a$, then some additional vehicles, $m_r$, should be rented. Then $m = m_r + m_a$.

The total fixed vehicle cost, $TC_f$, with a condition on the available number of vehicles is formulated as:

$$TC_f = f_a \, min \, \{ m_a, \sum_{j=1}^{n}\sum_{k=1}^{m} x_{0jk} \} + f_r \left( \sum_{j=1}^{n}\sum_{k=1}^{m} x_{0jk} - m_a \right)^{+}$$

where,

$$\left( \sum_{j=1}^{n}\sum_{k=1}^{m} x_{0jk} - m_a \right)^{+} = max \left\{ 0, \sum_{j=1}^{n}\sum_{k=1}^{m} x_{0jk} - m_a \right\},$$

$f_a$ and $f_r$ are the fixed costs of using one vehicle available at the depot and renting one vehicle, respectively, such that $f_r \geq f_a$. The total fixed cost of vehicles vs the number of vehicles used can be seen in Figure 3.1.

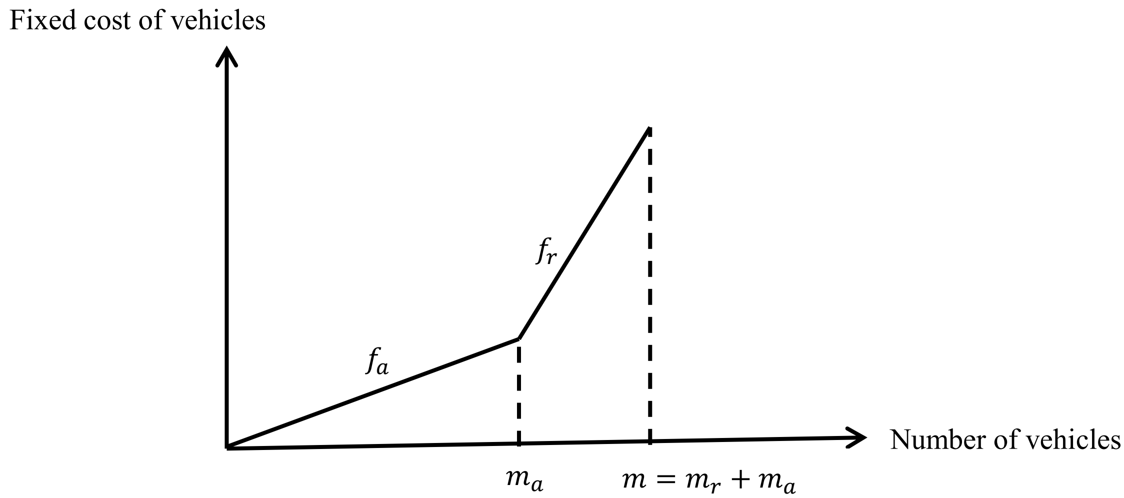



Figure 3.1. The total fixed cost of vehicles vs the number of vehicles

- **Penalty cost of late arrivals at customers**

In the case of 'soft' time windows, time window constraints can be violated, that is, they can be enlarged, but with a penalty cost. We use a variant of soft time windows in our model, such that early arrival is allowed at no cost while late arrival is allowed incurring a penalty cost.

In the early arrival of the vehicle, the vehicle must wait until the start time of the time window, and there is no penalty for the early arrival. When a vehicle arrives with an acceptable delay outside the time window, the product can still be delivered but with a penalty cost.

We consider the lateness cost as a penalty, in case a vehicle arrives at the customer later than the requested time window, due to two main reasons: some customers may be lost because the product is not available in the market, and some customers may refuse to buy the product due to the reduced shelf life. Then, some part of the demand cannot be met (lost sales), or met but not at the best market price.

Let the time range $[a_i, b_i]$ represent the preferred time window of customer $i$, and $a_i$ and $b_i$ be the start and end of the preferred delivery time window as imposed by the $i$th customer. Furthermore, let $e_i$ and $l_i$ denote respectively the earliest acceptable time for early arrival and the latest acceptable time for late arrival at customer $i$ such that $e_i \leq a_i$ and $l_i \geq b_i$. Therefore, the acceptable time ranges for early and late arrivals are within $[e_i, a_i]$ and $[b_i, l_i]$, respectively. A vehicle is allowed to arrive at customer $i$ only after time $e_i$ and before $l_i$, but allowed to start service after $a_i$ and before $l_i$. The vehicle is rejected when it arrives earlier than $e_i$ or later than $l_i$. When a vehicle arrives within the time ranges $[e_i, b_i]$ no penalty is incurred. When the vehicle arrives after $b_i$ but before $l_i$, a penalty that is related to the lateness time is considered as the revenue loss.

Let $c_b$ be the lateness penalty per vehicle per unit time, and the relationship between penalty costs and arrival time be given by a cost function denoted by $TC_{TW}$. When vehicle $k$ arrives at customer $i$ in $[a_i, b_i]$ and $[b_i, l_i]$ ranges, the arrival time and starting service time ($T_{ik}$) are the same; when vehicle $k$ arrives at customer $i$ in the $[b_i, l_i]$ range, a penalty of lateness arises after $b_i$, that can be estimated as: $c_b\ (T_{ik} - b_i)$.

When vehicle $k$ arrives at customer $i$ in $[e_i, a_i]$, no penalty is incurred for earliness, but the arrival and starting service times are different, in which case, $T_{ik}$ is the beginning of time window as imposed by customer $i$ ($T_{ik} = a_i$). The relationship between penalty costs and $T_{ik}$ at customer $i$ is illustrated in Figure 3.2.

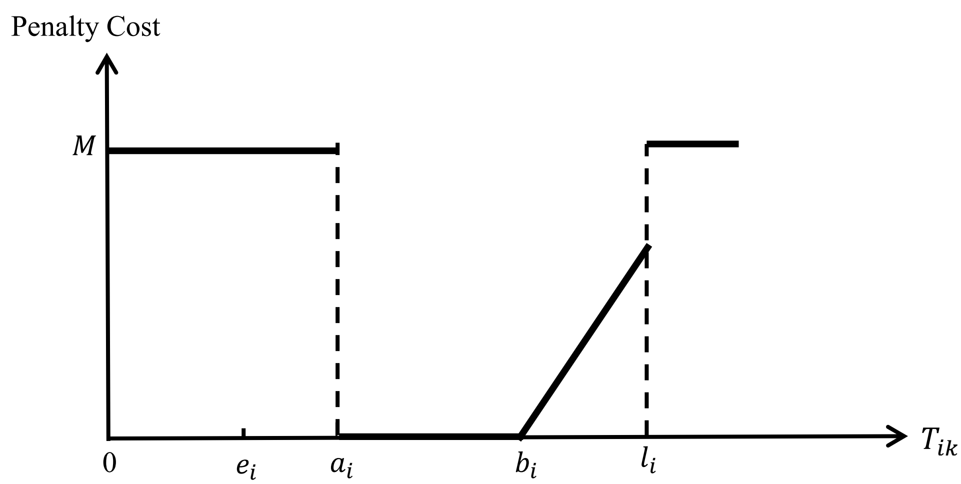



Figure 3.2. The relationship between penalty costs and $T_{ik}$ at customer $i$

We consider the penalty cost for violating the time window specified by customer $i$ as the revenue lost due to the late arrival time of the vehicle. The total cost for late arrival $TC_{TW}$ can be written as:

$$TC_{TW} = \sum_{i \in N} \sum_{\substack{k \in K \\ T_{ik} \leq l_i}} c_b \, (T_{ik} - b_i)^+ \; + \sum_{i \in N} \sum_{\substack{j \in \acute{N} \\ i \neq j}} \sum_{k \in K} M \, (a_i x_{ijk} - T_{ik})^+,$$

where

$$(a_i x_{ijk} - T_{ik})^+ = max\,\{0, a_i - T_{ik}\}$$

$$(T_{ik} - b_i)^+ = max\,\{0, T_{ik} - b_i\},$$

$$T_{jk} \geq \max\{T_{ik}, a_i\} + s_i + t_{ij} - M\left(1 - x_{ijk}\right), \quad i = 0,1,2, \dots, n; \quad j = 1,2, \dots, n,$$

$$k = 1,2, \dots, m, \quad i \neq j, \text{ and}$$

$$e_j \sum_{\substack{i=0 \\ i \neq j}}^{n} x_{ijk} \leq \; T_{jk} \leq \; l_j \sum_{\substack{i=0 \\ i \neq j}}^{n} x_{ijk}\,, \quad j = 1,2, \dots, n; \quad k = 1,2, \dots, m.$$

The time window constraint for $T_{jk}$ guarantees that when arrival time is outside the time range $[e_i\,, l_i]$, customer *i* refuses to receive the product. Thus, the formulation avoids an arrival time earlier than $e_i$ or later than $l_i$. Besides, when vehicle $k$ arrives within the time range $[a_i\,, b_i]$, the terms, $(a_i x_{ijk} - T_{ik})^+$ and $(T_{ik} - b_i)^+$, guarantee that $TC_{TW}$ turns out to be zero. Also, when vehicle $k$ arrives within the time range $[e_i\,, a_i]$, a very high cost $M$ on $(a_i x_{ijk} - T_{ik})^+$ guarantees that $T_{ik}$ has value equaling $a_i$.

- **Loss-of-quality cost**

We assume *t*=0 represents the time of the best condition (freshest) of product quality that diminishes during three stages. The first stage lasts from time *t*=0 to departure time from the depot, while the second stage lasts from vehicle departure time until arrival time at customer. Time window constraints are soft; therefore, the third stage quality reduction is from arrival time at the customer to the start of the time window in the early arrival of the vehicle. In the "arrival after $a_i$" case, the arrival time and starting service times are the same and customer does not experience the third stage. The quality loss in all three stages is defined as "1 minus the ratio of remaining shelf life to the shelf life" as:

$$1-\left(\frac{sl-\max\{T_{ik},a_i\}}{sl}\right), \qquad i=1,\dots,n;\;\; k=1,\dots,m$$

where $T_{ik}$ is the time at which vehicle $k$ starts servicing customer $i$, and $sl$ denotes the shelf life. As discussed before, in the case of early arrival, when the vehicle arrives before $a_i$ but after $e_i$, the delivery (starting service) time and arrival time for customer $i$ are different. The earliest acceptable delivery time for customer $i$ is $a_i$; so the vehicle must wait until time $a_i$ and service starting time of customer $i$ is $a_i$. The starting service time is $T_{ik}$ that should be equal to $a_i$. Hence, $M\,(a_i x_{ijk}-T_{ik})^+$ in the objective function guarantees that $T_{ik}$ equals $a_i$. In the case of late arrival, when the vehicle arrives after $b_i$ but before $l_i$, the resulting lateness affects both quality level and revenue. In this case, the servicing time for customer $i$ is $T_{ik}$, because, in the interval later than $b_i$ and earlier than $l_i$, delivery is done as soon as the vehicle arrives at the customer. Figure 3.3 illustrates the quality loss over time, starting at $t$=0 at the 100% quality level.

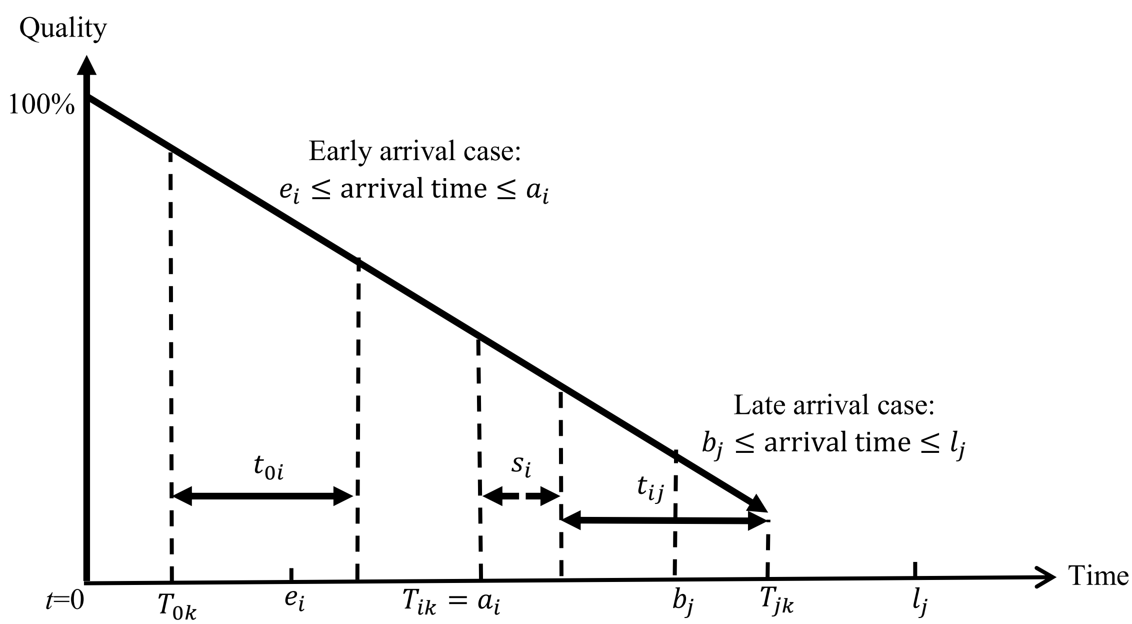



Figure 3.3. Quality loss of the product over time

The loss of quality level corresponds to a loss of revenue. The probability that perishable food can be sold, which is defined by the quality level, depends on the time between service delivery time and expiration date (remaining lifetime of the product).

Thus, the total quality loss cost, $TC_q$, is expressed as:

$$TC_q = \sum_{i \in N} \left(1 - \frac{sl - max\,\{T_{ik}, a_i\}}{sl}\right) p\, d_i$$

Defining $\frac{sl - \max\{T_{ik}, a_i\}}{sl}$ as a variable $q_i$ that denotes the quality level upon delivery at customer $i$, the expression for $TC_q$ above can be revised as:

$$TC_q = \sum_{i \in N} (1 - q_i) p\, d_i\,,$$

where $\quad q_i \leq \dfrac{sl - \max\{T_{ik}, a_i\}}{sl}$ for $i = 1, \dots, n;\;\; k = 1, \dots, m.$

Then, the objective function, $TC$, can be stated as the sum of the four cost elements discussed above:

$$TC = TC_{tr} + TC_f + TC_{TW} + TC_q$$

In the remainder of this chapter we present the mathematical formulations for the VRPSPD-STW-P.

### 3.2.2. Nonlinear formulation for VRPSPD-STW-P

We first discuss the non-linear formulation of the model for VRPSPD-STW-P and then linearize the model in section 3.2.3.

$$\text{minimize} \quad c \sum_{i \in \acute{N}} \sum_{\substack{j \in \acute{N} \\ i \neq j}} \sum_{k \in K} w_{ij} x_{ijk}$$

$$+ f_a \, min \, \{ m_a, \sum_{j=1}^{n} \sum_{k=1}^{m} x_{0jk} \} + f_r \left( \sum_{j=1}^{n} \sum_{k=1}^{m} x_{0jk} - m_a \right)^{+}$$

$$+ \sum_{i \in N} (1 - q_i) p \, d_i$$

$$+ \sum_{i \in N} \sum_{k \in K} c_b \, (T_{ik} - b_i)^{+} + \sum_{i \in N} \sum_{\substack{j \in \acute{N} \\ i \neq j}} \sum_{k \in K} M \, (a_i x_{ijk} - T_{ik})^{+} \qquad (1)$$

Subject to:

$$\sum_{\substack{i=0 \\ i \neq j}}^{n} \sum_{k=1}^{m} x_{ijk} = 1 , \qquad j = 1,2, \dots , n \qquad (2)$$

$$\sum_{\substack{j=0 \\ i \neq j}}^{n} \sum_{k=1}^{m} x_{ijk} = 1 , \qquad i = 1,2, \dots , n \qquad (3)$$

$$\sum_{\substack{j=0 \\ i \neq j}}^{n} x_{ijk} = \sum_{\substack{j=0 \\ i \neq j}}^{n} x_{jik} , \qquad i = 1,2, \dots , n; \quad k = 1,2, \dots , m \qquad (4)$$

$$\sum_{j=1}^{n} x_{0jk} \leq 1 , \qquad k = 1,2, \dots , m \qquad (5)$$

$$\sum_{i=0}^{n} \sum_{\substack{j=0 \\ i \neq j}}^{n} w_{ij} x_{ijk} \leq L , \qquad k = 1,2, \dots , m \qquad (6)$$

$$y_{ijk} \leq x_{ijk} \, Q , \qquad i = 0,1,2, \dots , n; \; j = 0,1,2, \dots , n; \; k = 1,2, \dots , m; \; i \neq j \qquad (7)$$

$$\sum_{j=1}^{n} y_{0jk} = \sum_{i=0}^{n} \sum_{\substack{j=0 \\ i \neq j}}^{n} x_{ijk}\, d_j\,, \quad k = 1,2,\dots,m \tag{8}$$

$$\sum_{\substack{i=0 \\ i \neq j}}^{n} y_{ijk} + \left(z_j - d_j\right) \sum_{\substack{i=0 \\ i \neq j}}^{n} x_{ijk} = \sum_{\substack{i=0 \\ i \neq j}}^{n} y_{jik}\,, \qquad j = 1,2,\dots,n; \qquad k = 1,2,\dots,m \tag{9}$$

$$T_{jk} \geq \max\{T_{ik}, a_i\} + s_i + t_{ij} - M\left(1 - x_{ijk}\right)\,, \quad i = 0,1,2,\dots,n; \quad j = 1,2,\dots,n \quad k = 1,2,..,m; \quad i \neq j \tag{10}$$

$$e_j \sum_{\substack{i=0 \\ i \neq j}}^{n} x_{ijk} \leq T_{jk} \leq l_j \sum_{\substack{i=0 \\ i \neq j}}^{n} x_{ijk}\,, \qquad j = 1,2,\dots,n; \qquad k = 1,2,\dots,m \tag{11}$$

$$q_i \leq \frac{sl - \max\{T_{ik}, a_i\}}{sl}, \qquad i = 1,\dots,n; \qquad k = 1,\dots,m \tag{12}$$

$$ql_i \leq q_i \qquad i = 1,\dots,n \tag{13}$$

$$x_{ijk} \in \{0,1\},\ T_{ik} \geq 0, q_i \geq 0,\ y_{ijk} \geq 0 \qquad i = 0,1,2,\dots,n; \quad j = 1,2,\dots,n \quad k = 1,2,..,m; \quad i \neq j \tag{14}$$

It should be noted that, in the model formulation (1)-(14) above, the nonlinearities are with the objective function (1) and the constraints (10) and (12).

### 3.2.3. Linear formulation for VRPSPD-STW-P

In this section, we introduce new decision variables in the VRPSPD-TW-P model and transform the nonlinear model to a linear one.

Let,

$V_a$: number of owned vehicles that are used,

$$V_a = min\left\{ m_a, \sum_{j=1}^{n}\sum_{k=1}^{m} x_{0jk} \right\}$$

$V_r$: number of rented vehicles,

$$V_r = \left( \sum_{j=1}^{n}\sum_{k=1}^{m} x_{0jk} - m_a \right)^+$$

$T_{aik}$: earliness, that is the waiting time of vehicle *k* until the start of time window, $a_i$,

$$T_{aik} = (a_i x_{ijk} - T_{ik})^+$$

$T_{bik}$: lateness of vehicle *k* at customer $i$,

$$T_{bik} = (T_{ik} - b_i)^+$$

$T_{mik}$: service starting time of vehicle *k* at customer *i*,

$$T_{mik} = max\,(T_{ik}, a_i)$$

By using these new variables defined above, the quality level of the product at customer *i* can then be expressed as follows:

$$q_i \leq \frac{sl - T_{mik}}{sl}$$

Therefore, a linear model can be formulated, (1)-(21) as follows:

minimize
$$c \sum_{i \in \acute{N}} \sum_{\substack{j \in \acute{N} \\ i \neq j}} \sum_{k \in K} w_{ij} x_{ijk} + f_a V_a + f_r V_r$$

$$+ \sum_{i \in N} (1 - q_i) p\, d_i + \sum_{i \in N} \sum_{k \in K} c_b T_{bik} + \sum_{i \in N} \sum_{k \in K} M\, T_{aik} \qquad (1)$$

Subject to:

$$\sum_{\substack{i=0\\i\neq j}}^{n}\sum_{k=1}^{m} x_{ijk} = 1\,, \qquad j = 1,2,\dots,n \tag{2}$$

$$\sum_{\substack{j=0\\i\neq j}}^{n}\sum_{k=1}^{m} x_{ijk} = 1\,, \qquad i = 1,2,\dots,n \tag{3}$$

$$\sum_{\substack{j=0\\i\neq j}}^{n} x_{ijk} = \sum_{\substack{j=0\\i\neq j}}^{n} x_{jik}\,, \qquad i = 1,2,\dots,n; \quad k = 1,2,\dots,m \tag{4}$$

$$\sum_{j=1}^{n} x_{0jk} \leq 1\,, \qquad k = 1,2,\dots,m \tag{5}$$

$$\sum_{i=0}^{n}\sum_{\substack{j=0\\i\neq j}}^{n} w_{ij}x_{ijk} \leq L\,, \qquad k = 1,2,\dots,m \tag{6}$$

$$y_{ijk} \leq x_{ijk}Q\,, \qquad i = 0,1,2,\dots,n; \quad j = 0,1,2,\dots,n, \quad k = 1,2,\dots,m; \quad i \neq j \tag{7}$$

$$\sum_{j=1}^{n} y_{0jk} = \sum_{i=0}^{n}\sum_{\substack{j=0\\i\neq j}}^{n} x_{ijk}\, d_j\,, \qquad k = 1,2,\dots,m \tag{8}$$

$$\sum_{\substack{i=0\\i\neq j}}^{n} y_{ijk} + \left(z_j - d_j\right)\sum_{\substack{i=0\\i\neq j}}^{n} x_{ijk} = \sum_{\substack{i=0\\i\neq j}}^{n} y_{jik}\,, \qquad j = 1,2,\dots,n; \quad k = 1,2,\dots,m \tag{9}$$

$$T_{jk} \geq T_{mik} + s_i + t_{ij} - M\left(1 - x_{ijk}\right)\,, \qquad i = 0,1,2,\dots,n; \quad j = 1,2,\dots,n, \quad k = 1,2,\dots,m; \quad i \neq j \tag{10}$$

$$e_j\sum_{\substack{i=0\\i\neq j}}^{n} x_{ijk} \leq T_{jk} \leq l_j\sum_{\substack{i=0\\i\neq j}}^{n} x_{ijk}\,, \qquad j = 1,2,\dots,n; \quad k = 1,2,\dots,m \tag{11}$$

$$q_i \leq \frac{sl - T_{mik}}{sl}, \qquad i = 1, \dots, n; \qquad k = 1, \dots, m \tag{12}$$

$$ql_i \leq q_i \qquad i = 1, \dots, n \tag{13}$$

$$x_{ijk} \in \{0,1\}, y_{ijk} \geq 0, T_{ik} \geq 0,\ q_i \geq 0, \qquad i = 0,1,2, \dots, n; \quad j = 1,2, \dots, n,$$
$$k = 1,2, \dots, m; \quad i \neq j \tag{14}$$

$$T_{mik} \geq T_{ik}, \qquad i = 1,2, \dots, n; \qquad k = 1,2, \dots, m \tag{15}$$

$$T_{mik} \geq a_i, \qquad i = 1,2, \dots, n; \qquad k = 1,2, \dots, m \tag{16}$$

$$T_{aik} \geq a_i x_{ijk} - T_{ik}, \qquad i = 1,2, \dots, n; j = 1,2, \dots, n;\ k = 1,2, \dots, m; \quad i \neq j \tag{17}$$

$$T_{bik} \geq T_{ik} - b_i, \qquad i = 1,2, \dots, n; \qquad k = 1,2, \dots, m \tag{18}$$

$$V_a \leq m_a, \tag{19}$$

$$\sum_{j=1}^{n} \sum_{k=1}^{m} x_{0jk} \leq V_a + V_r, \tag{20}$$

$$T_{aik} \geq 0, T_{bik} \geq 0, T_{mik} \geq 0,\ V \geq 0_a, V_r \geq 0,\ i = 0,1,2, \dots, n; k = 1,2, \dots, m \tag{21}$$

It is assumed that at $t$=0 all products are in their best quality level; that is $q_0 = 1$. The start and end times of the time window for the depot, and also the earliest and latest acceptable times for the depot are equal to zero; that is $a_0 = b_0 = e_0 = l_0 = 0$. As a result, we have:

$$T_{m0k} = T_{a0k} = T_{b0k} = 0.$$

The objective function (1) aims to minimize the total costs, consisting of five cost elements: transportation cost, fixed costs for the vehicles, penalty cost for late arrivals at customers. The last term (6th term) in the objective function is a very large cost term just to urge the service starting times to be at the starting times of the time windows if the vehicles arrive earlier than the time window. The quality cost function includes the loss-of-quality cost resulting from the storage time from $t$=0 until departure from the depot, travel time from the depot to the customer service time, and penalty cost for the lateness in late arrival cases.

Constraint sets (2) and (3) guarantee that each customer is visited exactly once by only one vehicle. Constraint set (4) states that every vehicle that arrives at a customer must leave the customer, and thus conservation of flow is satisfied. Besides, constraint set (4) means that all the vehicles start from the depot and terminate at the depot.

Constraint set (5) explains that each vehicle is used to serve at most one route and guarantees that the number of vehicles departing from the depot must be less than or equal to the total number of vehicles allowed. As a result, we have

$$\sum_{j=1}^{n}\sum_{k=1}^{m} x_{0jk} = \sum_{j=1}^{n}\sum_{k=1}^{m} x_{j0k} \leq m$$

Constraint set (6) guarantees that the total length of each route is limited by $L$. Constraint set (7) is the capacity constraint of each vehicle, enforcing that, if vehicle $k$ is serving customer $j$ after serving customer $i$ ($x_{ijk} = 1$), the corresponding load ($y_{ijk}$) must be less than or equal to the vehicle capacity ($Q$) after departing from customer $i$. Otherwise, $y_{ijk} = 0$ if $x_{ijk} = 0$. A result of this constraint set is that the total of all customer deliveries from the depot by a vehicle is less than or equal to the specified capacity. It can be stated for each vehicle as follows:

$$\sum_{j=1}^{n} y_{0jk} \leq Q \qquad\qquad k = 1,2, \dots, m$$

Constraint set (8) expresses that the total load of a vehicle at the departure from the depot must be equal to the sum of customer demands (deliveries) on its route. Constraint set (9) keeps track of the load carried by a vehicle as it visits the customers on its route either for delivery or pickup or both. It calculates the load of the vehicle after it serves customer $i$, and departs for customer $j$.

Constraint set (10) represents the precedence relation between two successive customers visited by a vehicle in terms of time.

This constraint set defines the earliest service starting time of a customer after its immediate preceding customer on the route. If vehicle $k$ serves customer $j$ after customer $i$ ($x_{ijk} = 1$), starting service time for customer $j$ must be greater than or equal to the sum of the service starting time for customer $i$, that is, $max(T_{ik}, a_i)$ in soft time window case, the service time at customer $i$, and transportation time from customer $i$ to customer $j$.

Otherwise, there is no relationship between the two service starting times of the two customers $i$ and $j$ ($x_{ijk} = 0$), and the constraint turns out to be redundant. Also, this constraint set guarantees the elimination of sub tours for a vehicle because it implies that the arrival time at a succeeding customer must be higher than the arrival time at the immediate past customer on the route that is not possible for all nodes in a sub tour.

Customers want their products to be available within a soft time window, and they need a specific time to be served. Constraint set (11) specifies the time window for arrival time and guarantees that, for $T_{jk}$ beyond time range $[e_j , l_j]$, customer $j$ refuses to receive the product, in which case it enforces $T_{jk} = 0$ and customer $j$ is not visited by vehicle $k$.

A lower bound is assumed for the quality level of the product upon delivery to the customer. In practice, customers prefer to receive their products at their highest quality level as long as it is possible, and they refuse products with a quality level under a specified limit. Hence the minimum preferred quality level of the customer is the lower bound on quality. If $ql_i$ is the lower bound of quality level imposed by customer $i$, then customer $i$ refuses to receive a product with a quality level in the range $[0, ql_i]$.

Constraint set (12) is the quality level expression when a customer receives the product.

Constraint set (13) presents lower bounds for the preferred quality levels specified by the customers. Constraint sets (14) express the domain of the decision.

Constraint sets (15) and (16) enforce to select $T_{mik} = max\,\{T_{ik}, a_i\}$. Constraint set (17) forces the service starting time at customer $i$ to be at the time window starting time $a_i$, if the vehicle arrives earlier. Constraint set (18) guarantees to select:

$$T_{bik} = max(0\,,\,T_{ik} - b_i).$$

Constraint sets (19) and (20) guarantee that the model selects

$$V_a = min\left\{m_a, \sum_{j=1}^{n}\sum_{k=1}^{m} x_{0jk}\right\} \text{ and } V_r = max\left\{0, \sum_{j=1}^{n}\sum_{k=1}^{m} x_{0jk} - m_a\right\}$$

in the feasible solutions. In other words, when the fleet of owned vehicles turns out to be insufficient, some vehicles can be rented.

Constraint sets (21) express the domain of the new decision variables for transforming the nonlinear model to a linear one.

In this chapter, we had an overview of the construction of the MILP for the VRPSPD-STW-P. We construct a type of routing problem to obtain optimal or near-optimal delivery routes, loads to be carried, and fleet dispatching and departure times from a central depot for delivering perishable food to customers at the highest possible quality. The proposed model is a MILP model representing a highly extended version of VRP, which is known to be NP-hard.

Since the VRPSPD-STW-P is an NP-hard problem that is to be solved at the operational level, we need to search for some heuristic approaches that provide the decision maker with optimal or near-optimal solutions in reasonable computational times.

In the next chapter, we present the solution methodology for our problem. We provide an exact solution approach and three heuristic approaches with an embedded metaheuristic algorithm to solve the VRPSPD-STW-P. We introduce the use of CPLEX optimization software as an exact solution tool, and we look for the details of the heuristic algorithms to obtain satisfactory solutions for this problem.

# CHAPTER 4

# SOLUTION METHODOLOGY

It is apprehensible that obtaining an optimal or near-optimal solution for the distribution process of perishable food products, where time is the most important factor, recommends the use of algorithms that can find good solutions in reasonable computational time.

In this chapter, we propose the exact solution approach and heuristic algorithms to solve the VRPSPD-STW-P. The VRPSPD-STW-P is a special case of the classical VRP. Since, in itself, the classical VRP is NP-hard and has high complexity (Lenstra and Rinnooy Kan, 1981), so that for practical purposes, heuristic approaches and the methods based on various metaheuristics have been widely used in solving VRP and its extensions.

We employ a clustering approach for decomposing the VRPSPD-STW-P into $m$ traveling salesman problem since solving these reduced-size problems is more straightforward than the main problem. Furthermore, we develop a genetic algorithm (GA) as a metaheuristic approach for improving the initial solution. We introduce three routing heuristics to generate solutions, obtain an initial solution for each cluster, and improve the initial solutions by the genetic algorithm. Our approach focuses on capacity and tour length constraints violation in the TSPSPD and time-window and quality constraints violation in the TSPSPD-STW-P for obtaining a solution to the original problem, VRPSPD-STW-P.

## 4.1. Exact solution approach

We adopt the strategy of using version 12.7.0.0 of IBM ILOG CPLEX Optimization Studio to run each of the test problems corresponding to the VRPSPD-STW-P linear formulation discussed in Chapter 3. The proposed algorithm by CPLEX is a branch-and-bound method.

## 4.2. Heuristic solution approach

In this section, we present a method with mainly two phases in succession for obtaining initial solutions for the problem and then improving the initial solutions. Considering the VRPSPD-STW-P as an mTSPSPD-STW-P, we obtain a solution for each TSPSPD-STW-P as follows: (i) randomly generating a giant tour of all nodes in the network, clustering the giant tour to $m$ groups (routes) for generating $m$ solutions of the mTSPSPD-STW-P, (ii) routing each cluster by three different routing heuristics and (iii) checking the feasibility of the $m$ routes for obtaining the initial feasible solutions.

Thus, in the first phase, we obtain initial feasible solutions that constitute the initial gene pool of the genetic algorithm of the second phase. In the second phase, we adopt a genetic algorithm (GA) for improving the initial feasible solutions and determining the best minimum solution. A basic GA starts with a population of candidate solutions that are generated in the first phase. In both phases, the feasibility of the constructed routes is checked in three steps.

The two phases of the solution approach to obtain a feasible solution for the VRPSPD-STW-P are explained below.

### Phase 1: Initial solution algorithm

The steps of generating initial feasible solutions for the VRPSPD-STW-P are detailed below.

(i) Clustering (grouping):

- Randomly generate a giant tour of all demand nodes in the network starting from and ending at the depot.
- Cluster the giant tour in *m* routes. Each cluster corresponds to a route, and each route is a generated solution (feasible or infeasible) for the TSPSPD-STW-P.

(ii) Routing:

- Form route sets for the *m* clusters by using one of the three routing heuristics.

(iii) Checking feasibility for each route $k$ , $k$={0,1,…,$m$-1}:

- Feasibility of TSPSPD for each route *k*: Check the feasibility in terms of distance and vehicle capacity constraints for route *k*. If it is feasible, check the quality and time window constraints.
- Feasibility of TSPSPD-STW-P for route *k*: Check the feasibility in terms of quality and time window constraints for route *k*. If it is feasible, then it a solution for TSPSPD-STW-P.
- Thus, the initial solution for the VRPSPD-STW-P is obtained by combining *m* feasible routes of mTSPSPD-STW-P.

**Phase 2: Improvement algorithm: Genetic algorithm (GA)**

The steps of the genetic algorithm to improve the initial solution for VRPSPD-STW-P are summarized below.

(i) Clustering (grouping):

- Generate new chromosomes by crossover and mutation operators. Decoding each chromosome presents the clusters directly.
- Each cluster corresponds to a route, and each route is a candidate solution for TSPSPD-STW-P.

(ii) Routing:

- Form route sets for $m$ clusters of the decoded chromosome, using one of the three routing heuristics.

(iii) Checking feasibility for each route $k$, $k=\{0,1,\ldots,m-1\}$:

- Feasibility of TSPSPD for each route $k$: Check the feasibility in terms of distance and vehicle capacity constraints for route $k$. If it is feasible, check the quality and time-window constraints.

- Feasibility of TSPSPD-STW-P for route $k$: Check the feasibility in terms of quality and time window constraints for route $k$. If it is feasible, then it is a solution for TSPSPD-STW-P.

- Thus, the GA solution for the VRPSPD-STW-P is obtained by combining $m$ feasible routes of mTSPSPD-STW-P.

For Step (iii), see Figure 4.1. below.

It should be noted that Steps (ii) and (iii) of Phase 1 and Phase 2, namely "routing" and "checking feasibility", are common for both phases of our solution methodology.

In the following subsections, we discuss the steps of the two phases. Firstly, the first step of Phase 1, clustering, is explained. Next, the second and third steps, "routing" and "checking feasibility" steps of both Phase 1 and Phase 2, are discussed. Finally, the genetic algorithm of Phase 2 is explained, which corresponds to the first step of Phase 2.

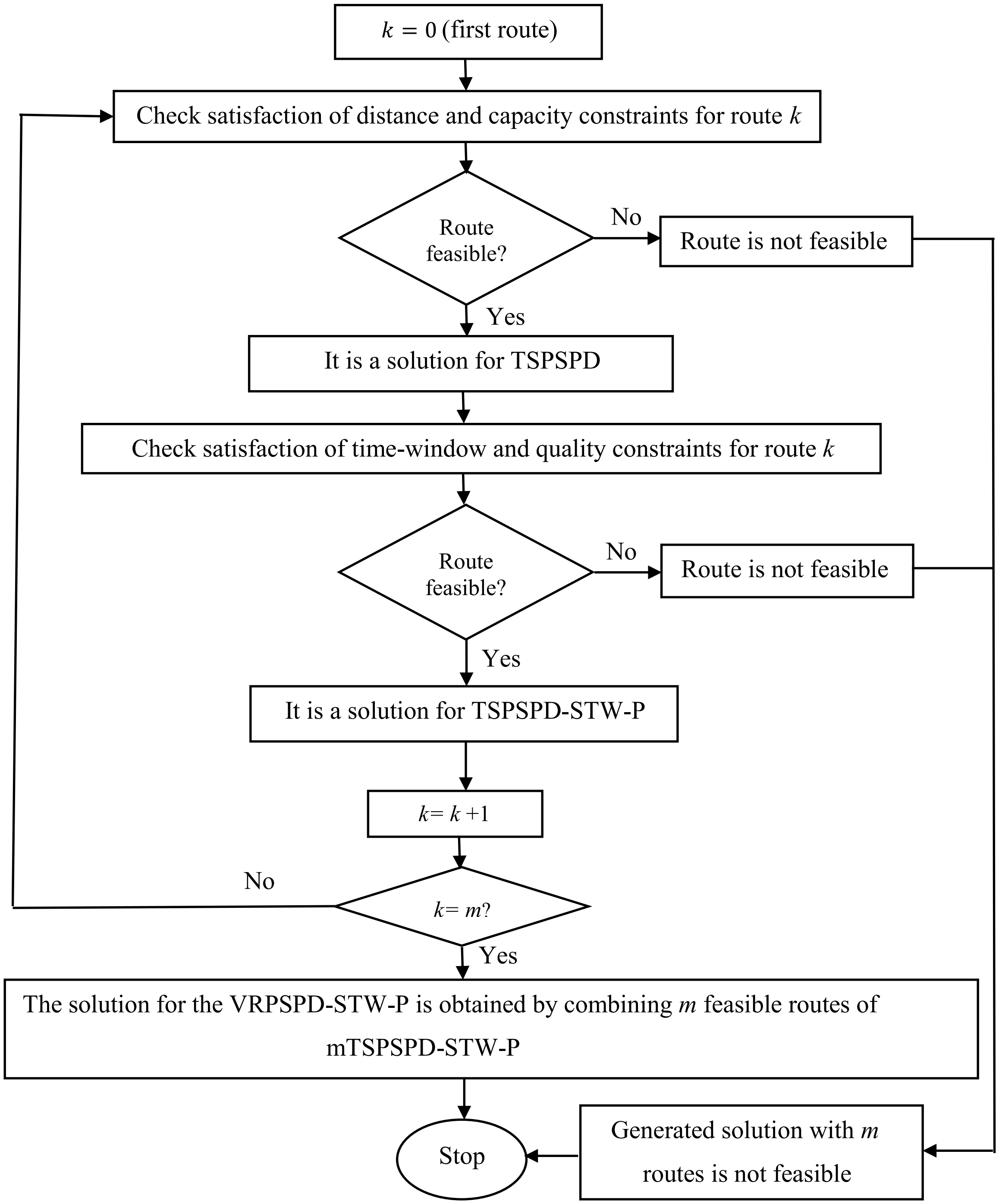



Figure 4.1. Step (iii): Feasibility checking

### 4.2.1. Step 1 of Phase 1: Clustering (grouping)

In the first step of the solution methodology and hence of Phase 1, to obtain initial feasible solutions, all nodes of the network are considered a giant tour. The initial giant tour is generated as a TSP tour without considering vehicle capacity, tour distance, and time window constraints.

Then, we partition them into $m$ clusters such that each cluster corresponds to a route for a vehicle and each route to a solution (feasible/infeasible) for TSPSPD-STW-P. If all routes are found to be feasible, these $m$ solutions turn out to be a feasible solution for VRPSPD-STW-P. This clustering approach reduces the problem to the mTSPSPD-STW-P, i.e., forms $m$ many smaller-sized TSPs, thus enabling the use of more straightforward solution procedures and obtaining a feasible solution in reasonable computing time for VRPSPD-STW-P.

In this step, a set of randomly generated giant tours is obtained, each tour representing a random routing of nodes. We consider the pool size to be $g$; therefore, $g$ many giant tours are generated, i.e., $g$ many solutions some of which may turn out to be infeasible for the original problem VRPSPD-STW-P.

Suppose $NODES$ = […] is a randomly generated matrix. It is an array with $n$ elements and represents the random routing sequence of nodes in the initial giant tour. The number of nodes in each route $k$, $k=\{0,1,\ldots,m-1\}$, is shown by the array $ar$ = […], which has $m$ positions, $ar[0]$, $ar[1]$,..., $ar[m-1]$, where $ar[i]$ shows the number of nodes assigned to route $i$ and are served by vehicle $i$.

The $Route_k$ is the route $k$ set and presents the nodes that are in route $k$ and served by vehicle $k$ in the order as stated in the set. Set $N_k$ ($k=\{0,1,\ldots,m-1\}$) denotes the nodes in $Route_k$ or cluster $k$ and $|N_k|$ the cardinality of route $k$. The vehicles, which serve the corresponding nodes in the same position in the *NODES* set, are expressed by the vehicle set *vehicle*.

**Example:** Suppose *NODES* shows the order of nodes in a giant tour that is randomly generated for a problem instance with $n$=10 nodes and $m$=3 vehicles (or routes).

$$NODES=\{5,8,7,2,6,10,1,4,3,9\}$$

The result of dividing this giant tour into three routes (clusters) is the number of nodes in each route. Say, for example, $ar[0]=2$, $ar[1]=4$, and $ar[2]=4$ are the number of nodes assigned to routes 0, 1 and 2. Therefore, the number of nodes that should be visited by vehicles 0, 1, and 2 are 2, 4, and 4, respectively. Then, the vehicle set of the problem can be expressed as follows:

$$vehicle=\{\ 0\ 0\ 1\ 1\ 1\ 1\ 2\ 2\ 2\ 2\ \}$$

This set presents that nodes 5 and 8 are served by vehicle 0; and nodes 7, 2, 6, and 10 by vehicle 1; and nodes 1, 4, 3, and 9 by vehicle 2. Then, route sets are:

$$Route_0=[\ 5\ 8\ ]$$
$$Route_1=[\ 7\ 2\ 6\ 10\ ]$$
$$Route_2=[\ 1\ 4\ 3\ 9\ ]$$

Therefore, $N_0=\{5,8\}$, $N_1=\{7,2,6,10\}$, and $N_2=\{1,4,3,9\}$, including the depot in each route, $\acute{N_k} = N_k \cup \{0\}$. As discussed above, in the route that is served by vehicle $k$, the traveling sequence of the customer nodes is determined based on the order of nodes in each route set. For example, the node in position 0, which is node 7, in $Route_1$ is served before node in position 1, which is node 2; node in position 1, which is node 2, is served before node in position 2, which is node 6; node in position 2, which is node 6, is served before node in position 3, which is node 10. Therefore, the traveling sequence is based on the order of customers in $Route_k$ set.

The pseudocode of clustering for a generated giant tour is given in the following algorithm.

**Algorithm: Clustering**

1: **Begin**

2: Generate a routing sequence of all nodes randomly, and call it as a giant tour.

3: Divide the giant tour randomly into $m$ routes.

4: Set $N_k$ and $|N_k|$.

5: **End**

### 4.2.2. Step 2 of Phases 1 and 2: Routing

The routing sequence of the customer nodes in route $k$ ($k$={0,1,…,$m$-1}) that are served by vehicle $k$ is defined by their order in $Route_k$ set. Routing for nodes in each cluster is used in both solution phases, Phase 1 and Phase 2; that is, the initial solution generation and the improvement phase, genetic algorithm, respectively. Once the clusters of nodes are formed, we use three heuristic methods for routing in each cluster. Figure 4.2 shows these three heuristics for routing in each cluster.

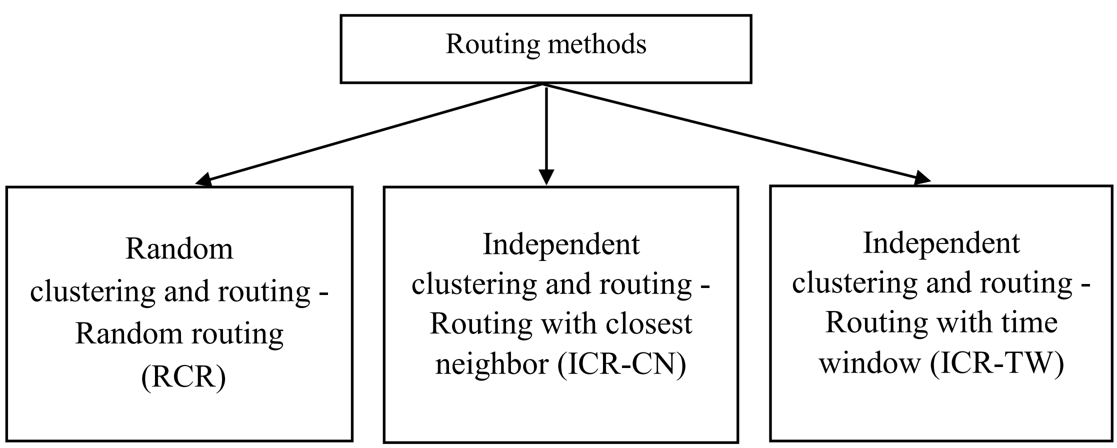



Figure 4.2. Heuristic methods for routing

#### 4.2.2.1. Random clustering and routing (RCR)

In this method, the routing of nodes is the same as the randomly generated nodes set *NODES* (giant tour). $Route_k$ set represents the random routing sequence of nodes in route $k$ with the same sequence as in *NODES*. Therefore, the clustering and routing are obtained by means of a random strategy. The pseudocode of this method, RCR, is illustrated in the following algorithm.

---

**Algorithm: Random clustering and routing (RCR)**

---

1: **Begin**

2: Generate a routing sequence of all nodes randomly, and call it as a giant tour.

3: Divide the giant tour randomly into $m$ routes.

4: Keep routing in each cluster as the same as the order of nodes in the giant tour.

5: Set $N_k$ and $|N_k|$.

6: **End**

---

#### 4.2.2.2. Independent clustering and routing with the closest neighbor heuristic (ICR-CN)

In this method, we apply the strategy of group-first, route-second. After grouping the nodes in clusters, we implement the closest neighbor heuristic to obtain routing in each cluster. The pseudocode of the ICR-CN heuristic is given below. The $w_{ij}$ in the algorithm means travel distance from customer $i$ to customer $j$.

---

**Algorithm: Independent clustering and routing with closest neighbor heuristic (ICR-CN)**

---

1: **Begin**

2: Generate a routing sequence of all nodes randomly, and call it as a giant tour.

3: Divide the giant tour randomly into $m$ clusters (routes).

4: Let $N_k$ = set of nodes in route $k$, and $|N_k|$ the cardinality of route $k$.

5: **for** routes $k$=1 to $m$ do

6: Consider the depot as the beginning of route $k$.

7: **for** nodes in positions $i$=1 to | $N_k$ | do

8: Find $w_{0N_k[i]}$.

9: **end for**

10: Sort all nodes in cluster $k$ in ascending order of $w_{0N_k[i]}$.

11: Assign node with minimum $w_{0N_k\,[i]}$ to $N_k$ [1].

12: **for** nodes in positions $i$=1 to | $N_k$ |-1 do

13: **for** nodes in positions $j$=$i$+1 to | $N_k$ | do

14: Find $w_{iN_k\,[i]}$.

15: **end for**

16: Assign node with minimum $w_{iN_k\,[j]}$ to $N_k$ [$i$+1].

17: **end for**

18: Return to the depot.

19: **end for**

20: **End**

---

Lines 1-3 initialize the algorithm by generating the giant tour and clustering. In line 4, the algorithm starts routing nodes in all $m$ clusters based on the ICR-CN heuristic. Line 5 enumerates the routes of the generated giant tour. In lines 6-9, the depot is denoted as the starting point for route $k$, and the distances between the depot and each node in route $k$ are determined, and then in lines 10 and 11, the closest node to the depot is determined, which turns out to be the first traveling point from the depot. In lines 12-17, the nearest unvisited node to the last node added to the route is found, and the routing sequence of nodes in the route is determined by the closest neighbor heuristic. When all nodes are routed, and there remains no unvisited node in the route $k$, the vehicle returns to the depot from the last node of the route in line 18. The algorithm is repeated until all clusters of the giant tour are routed.

### 4.2.2.3. Independent clustering and routing with time window heuristic (ICR-TW)

In this method, we apply the strategy of group-first, route-second as in ICR-CN. After grouping the nodes in clusters, we implement a time window heuristic to determine the routing of each cluster. The pseudocode of the ICR-TW heuristic is given in the following algorithm. Again, it should be noted here that $e_i$ and $l_i$ are the earliest and latest acceptable arrival times at node *i*.

---

**Algorithm: Independent clustering and routing with time window heuristic (ICR-TW)**

---

**Begin**

Generate a routing sequence of all nodes randomly, and call it as a giant tour.

Divide the giant tour randomly into *m* clusters (routes).

Let $N_k$ = set of nodes in route *k*, and $|N_k|$ the cardinality of route *k*.

**for** routes $k$=1 to $m$ do

Consider the depot as the beginning of route *k*.

Find the latest acceptable arrival times ( $l_{N_k[i]}$ ) for all nodes in route *k*.

Sort all nodes in route *k* in ascending order of their $l_{N_k[i]}$ .

**for** nodes in positions $i$=1 to $|N_k|$-1 do

**for** nodes in positions $j$=$i$+1 to $|N_k|$ do

**if** $l_{N_k[i]} < l_{N_k[j]}$ **then**

Assign node in position *i* to $N_k[i]$.

**else if** $l_{N_k[i]} = l_{N_k[j]}$ and $l_{N_k[i]} - e_{N_k[i]} < l_{N_k[j]} - e_{N_k[j]}$ **then**

Assign node in position *i* to $N_k[i]$.

**else**

Assign node in position *j* to $N_k[i]$.

**end if**

**end for**

**end for**

Return to the depot.

**end for**

**End**

---

Lines 1-3 initialize the algorithm by generating the giant tour and clustering. In line 4, the algorithm starts routing nodes in all *m* clusters based on the ICR-TW heuristic. Line 5 enumerates the routes of the generated giant tour. In lines 6-8, the depot is denoted as the starting point for route *k*, and nodes in route *k* are sorted by the ascending order of their latest acceptable arrival times. Lines 9-19 correspond to the main part of the heuristic method since the routing sequence of nodes in a route is determined. This main part of the algorithm consists of two steps: for all nodes in $N_{\mathrm{k}}$, position the node in position *i* before *j* if $l_i < l_j$; otherwise, if $l_i = l_j$ and $l_i – e_i < l_j – e_j$, then position the node in position *i* before *j*; otherwise, position the node in position *j* before *i*. When all nodes are routed, and there remains no unvisited node in route *k*, the vehicle returns to the depot from the last node of the route in line 20. The algorithm is repeated until routing of all clusters is completed.

### 4.2.3. Step 3 of Phases 1 and 2: Checking for feasibility

After clustering, routing the clusters, and thus forming *m* routes, we should check for the feasibility of *m* routes under the constraints of simultaneous pickup and delivery with time window and quality requirements.

First, the route *k*, *k*={0,1,…,*m*-1}, is checked for simultaneous pickup and delivery requirements only. So for route *k,* we end up with a TSPSPD. In other words, route *k* is a candidate feasible solution for TSPSPD.

In the case of feasibility for route *k* under pickup and delivery requirements, then, it is checked for feasibility in terms of the time windows of the customers on the route and the quality requirements of the customers on the route. In this step of feasibility checks, we are addressing the TSPSPD-STW-P for route *k*.

If each route $k$ of $m$ routes turns out to be feasible, considering all $m$ feasible routes, the problem addressed, then, is mTSPSPD-STW-P, the solution of which becomes a solution for the original problem VRPSPD-STW-P.

If any one of $m$ routes becomes infeasible, then these m routes (of the giant tour) are all discarded.

The details of Step 3 are given in the following subsections.

#### 4.2.3.1. Obtaining solutions for TSPSPD

To find out whether route $k$, $k$={0,1,…,$m$-1}, is a feasible solution for TSPSPD, we proceed as follows:

Let $z_{cvk}$ for $k$={0,1,…,$m$-1} denote the cost of route $k$, consisting of the transportation costs and fixed costs of the vehicle used. The TSPSPD below is solved given route $k$, considering the delivery and pickup quantities of all customers on route $k$. If the solution for TSPSPD for route $k$ turns out to be feasible, we keep the objective function value, $z_{cvk}$, and go on with route $k$ for checking feasibility in terms of the time windows and quality requirements of the customers on the route.

As a result, for the nodes that are served by vehicle $k$ in the $k$th cluster ($Route_k$), the cost of each route can be obtained by the corresponding problem of each cluster.

- **TSPSPD model**

TSPSPD formulation is presented below. It should be noted here that $x_{ijk}$ decision variables are fixed based on the order of customers served in route $k$.

Parameters and decision variables are defined for the mathematical model of TSPSPD for each route as follows:

**Parameters:**

$Q$: vehicle capacity

$w_{ij}$: travel distance from customer $i$ to customer $j$

$d_i$: demand quantity of customer $i$

$z_i$: pickup quantity of customer $i$

$c$: transportation cost per unit distance traveled by an owned/rented vehicle

$L$: Maximum length of each route

$f_a$: fixed cost for a vehicle available at the depot

$f_r$: fixed cost for a rented vehicle

$m_a$: number of vehicles owned and available at the depot

$m_r$: number of rented vehicles at the depot

$m$: minimal number of vehicles for a feasible solution

$x_{ijk}$: equals 1 if arc $(i, j)$ is traveled by vehicle $k$, 0 otherwise

**Decision Variables:**

$y_{ijk}$: load of vehicle $k$ when vehicle $k$ departs from customer $i$ toward customer $j$

The TSPSPD model is formulated below:

Compute $$z_{cvk} = c \sum_{i \in N'_k} \sum_{\substack{j \in N'_k \\ i \neq j}} w_{ij} x_{ijk} + f_a \min((m_a - k)^+, 1)$$

$$+ f_r \min((k - m_a + 1)^+, 1) \qquad (1)$$

Subject to:

$$\sum_{\substack{i \in N'_k \\ i \neq j}} x_{ijk} = 1, \qquad j \in N_k \qquad (2)$$

$$\sum_{\substack{j \in \acute{N}_k \\ i \neq j}} x_{ijk} = 1 , \qquad i \in N_k \tag{3}$$

$$\sum_{\substack{j \in \acute{N}_k \\ i \neq j}} x_{ijk} = \sum_{\substack{j \in \acute{N}_k \\ i \neq j}} x_{jik} , \qquad i \in N_k \tag{4}$$

$$\sum_{j \in N_k} x_{0jk} \leq 1 , \tag{5}$$

$$\sum_{i \in \acute{N}_k} \sum_{\substack{j \in \acute{N}_k \\ i \neq j}} w_{ij} x_{ijk} \leq L , \tag{6}$$

$$y_{ijk} \leq x_{ijk} Q , \qquad i \in \acute{N}_k , \quad j \in \acute{N}_k \qquad i \neq j \tag{7}$$

$$\sum_{j \in N_k} y_{0jk} = \sum_{i \in \acute{N}_k} \sum_{\substack{j \in \acute{N}_k \\ i \neq j}} x_{ijk} \, d_j , \tag{8}$$

$$\sum_{\substack{i \in \acute{N}_k \\ i \neq j}} y_{ijk} + (z_j - d_j) \sum_{\substack{i \in \acute{N}_k \\ i \neq j}} x_{ijk} = \sum_{\substack{i \in \acute{N}_k \\ i \neq j}} y_{jik} , \qquad j \in N_k \tag{9}$$

$$y_{ijk} \geq 0 \qquad i \in \acute{N}_k , \quad j \in N_k \qquad i \neq j \tag{10}$$

The results of the feasible TSPSPD solutions (i.e., feasible routes) and summation of the objective function values of all *m* routes are returned to the main problem as a performance criterion for evaluating the current clustering result.

Let us define the cost terms in (1) above as:

Travelling cost: $$z_{ck} = c \sum_{i \in N'_k} \sum_{\substack{j \in N'_k \\ i \neq j}} w_{ij} x_{ijk}$$

Vehicle cost: $$z_{vk} = f_a \min((m_a - k)^+, 1) + f_r \min((k - m_a + 1)^+, 1)$$

Then, the objective function value for *Route*$_k$ (TSPSPD solution with route $k$) is defined as:

$$z_{cvk} = z_{ck} + z_{vk}$$

It should be noted again here that VRPSPD is formed by mTSPSPD, and the objective function value of the problem for all $m$ routes becomes:

$$z_{\text{mTSPSPD}} = \sum_{k \in K} z_{cvk} \quad \text{where } K = \{0,1,\dots,m-1\}.$$

The solution for the TSPSPD formulation (1)-(10) displays whether *Route*$_k$ is a feasible route considering the simultaneous pickup and delivery requirements of the nodes on the route without exceeding the vehicle $k$ capacity and route length limitation. The pseudocode and the steps of the solution algorithm for each route $k$, $k=\{0,1,\dots,m-1\}$, are illustrated below.

**Algorithm: Solution approach for route *k* (TSPSPD)**

1: **Begin**

2: Set $z_{ck}=0$, $z_{vk}=0$, $z_{cvk}=0$, $W=0$, $Y=0$, consider $N_k[0]$ as depot.

3: Set $W = w_{N_k[0]N_k[1]} + w_{N_k[|N_k|]N_k[0]}$

4: **for** nodes in positions $i$=1 to $|N_k|$-1 do

5: Determine the distance between two sequential nodes ($w_{N_k[i]N_k[i+1]}$).

6: Set $W = W + w_{N_k[i]N_k[i+1]}$

7: **end for**

8: **if** $W \leq L$ **then**

9: **for** $i$=1 to $|N_k|$ do

10: Determine delivery demands of all nodes ($d_{N_k[i]}$).

11: Set $Y = Y + d_{N_k[i]}$

12: **end for**

13: Set $y_{N_k[0]N_k[1]} = Y$

14: **for** nodes in positions $i$=1 to $|N_k|$-1 do

15: Find load of the vehicle between two successive nodes ($y_{N_k[i]N_k[i+1]}$).

16: Set $y_{N_k[i]N_k[i+1]} = y_{N_k[i-1]N_k[i]} + z_{N_k[i]} - d_{N_k[i]}$

17: **if** $y_{N_k[i]N_k[i+1]} \leq Q$ **then**

18: Set $z_{ck} = z_{ck} + c\, w_{N_k[i]N_k[i+1]}$

19: **else**

20: "Route is infeasible", and go to 38

21: **end if**

22: **end for**

23: Set $y_{N_k[|N_k|]N_k[0]} = y_{N_k[|N_k|-1]N_k[|N_k|]} + z_{N_k[|N_k|]} - d_{N_k[|N_k|]}$

24: **if** $Y \leq Q$ and $y_{N_k[|N_k|]N_k[0]} \leq Q$ **then**

25: Set $z_{ck} = z_{ck} + c\,(w_{N_k[0]N_k[1]} + w_{N_k[|N_k|]N_k[0]})$

26: **if** $k \leq m_a$ **then**

27: Set $z_{vk} = f_a$

28: **else**

29: Set $z_{vk} = f_r$

30: **end if**

31: Set $z_{cvk} = z_{ck} + z_{vk}$

32: **else**

33: "Route is infeasible"

34: **end if**

35: **else**

36: "Route is infeasible"

37: **end if**

38: **End**

In the pseudocode, lines 1-2 initialize the algorithm with the starting values of cost parameters and sets. In line 3, the sum of the distances from the depot to the first node and from the last node to the depot in route $k$ is obtained. Line 4 enumerates the demand nodes in route $k$. In lines 5-7, the total distance traveled by the vehicle in route $k$ is calculated. The distance traveled limitation is checked in line 8. If it is satisfied, then the algorithm goes on checking the feasibility of vehicle capacity in lines 9-24. Otherwise, the route is infeasible, and the algorithm is terminated by reporting "Route is infeasible". Lines 9-24 find the total of all customer deliveries from the depot by vehicle $k$, the load of vehicle $k$ in the traversed arcs on route $k$, and check the feasibility of capacity for the vehicle load on each arc. Lines 17 and 24 determine if the load of the vehicle in the traversed arcs of route $k$ is less than or equal to the vehicle capacity $Q$, then route $k$ is said to be feasible and a solution for the TSPSPD. Otherwise, the route is infeasible, and the algorithm is terminated by reporting "Route is infeasible" and goes to line 38. The traveling route cost is determined in lines 18 and 25, and the fixed vehicle cost is calculated in lines 26-30, and the total cost of the TSPSPD is reported in line 31. It is to be noted here that each feasible route $k$ of any giant tour at this point is to be checked whether it is feasible in terms of soft time windows and quality restrictions, which means that TSPSPD-STW-P is now has to be formulated as illustrated in the following section.

#### 4.2.3.2. Obtaining solutions for TSPSPD-STW-P and the main problem

The solution of the TSPSPD for route $k$, i.e., decision variables' values, route information, and the objective function value (traveling and vehicle costs), are passed to the TSPSPD-STW-P formulation and evaluated for feasibility for the time window and quality limitations.

- **TSPSPD-STW-P model**

Parameters and decision variables are defined for the mathematical model of the TSPSPD-STW-P for each route $k$ as follows.

**Parameters:**

$s_i$: service time at customer $i$

$t_{ij}$: travel time from customer $i$ and customer $j$

$a_i$: start of time window for arriving at customer $i$

$b_i$: end of time window for arriving at customer $i$

$e_i$: earliest acceptable time for arrival at customer $i$

$l_i$: latest acceptable time for arrival at customer $i$

$p$: selling price of one-unit food product at the depot

$c_b$: lateness penalty cost per vehicle per unit of time

$sl$: shelf-life of perishable food product

$ql_i$: lowest quality level acceptable by customer $i$

$x_{ijk}$: equals 1 if arc $(i, j)$ is traveled by vehicle $k$, 0 otherwise

**Decision Variables:**

$T_{ik}$: time when vehicle $k$ starts servicing customer $i$

$q_i$: final quality level of the product when service starts at customer $i$

$z_k$ denotes the total cost for $Route_k$ , and it is the summation of objective function values resulting from TSPSPD and TSPSPD-STW-P, consisting of the transportation costs and fixed costs of the vehicle used in TSPSPD, and quality and time window related costs in TSPSPD-STW-P.

The TSPSPD-STW-P for $Route_k$ is formulated as follows:

$$Compute \qquad z_k = z_{ck} + z_{vk} + \sum_{i \in N_k} (1 - q_i) p\, d_i$$

$$+ \sum_{i \in N_k} c_b \, (T_{ik} - b_i)^+ + \sum_{i \in N_k} \sum_{\substack{j \in \acute{N}_k \\ i \neq j}} M \, (a_i x_{ijk} - T_{ik})^+ \qquad (1)$$

Subject to:

$$T_{jk} \geq \max\{T_{ik}, a_i\} + s_i + t_{ij} \;, \quad i \in N_k \;, \quad j \in \acute{N}_k \quad i \neq j \qquad (11)$$

$$e_j \sum_{\substack{i \in \acute{N}_k \\ i \neq j}}^{n} x_{ijk} \leq T_{jk} \leq l_j \sum_{\substack{i \in \acute{N}_k \\ i \neq j}}^{n} x_{ijk} \;, \qquad j \in N_k \qquad (12)$$

$$q_i \leq \frac{sl - \max\{T_{ik}, a_i\}}{sl}, \qquad i \in N_k \qquad (13)$$

$$ql_i \leq q_i \qquad i \in N_k \qquad (14)$$

$$T_{ik} \geq 0, q_i \geq 0, \qquad i \in \acute{N}_k \;, \quad j \in N_k \quad i \neq j \qquad (15)$$

Let us define:

Loss-of-quality cost as

$$z_{qk} = \sum_{i \in N_k} (1 - q_i) p\, d_i$$

and penalty cost for time window violation (lateness) as

$$z_{bk} = \sum_{i \in N_k} c_b \, (T_{ik} - b_i)^+.$$

Then the total cost for *Route*$_k$ is

$$z_k = z_{ck} + z_{vk} + z_{qk} + z_{bk}.$$

If all *m* routes of the giant tour are found to be feasible at this point, then the summation of the objective function values obtained by all *m* feasible solutions in the current giant tour generation is the objective function value of the main problem, VRPSPD-STW-P, and calculated as:

$$z = \sum_{k \in K} z_k \qquad\qquad K = \{0,1,\ldots,m-1\}$$

This feasible solution is stored as a satisfactory solution for the main problem and the algorithm then goes on with another new generation.

The pseudocode of the solution algorithm for the problem for route *k* (TSPSPD-STW-P) is given below:

---

**Algorithm: Solution approach for route *k* (TSPSPD-STW-P)**

---

1: **Begin**

2: Set $z_{ck}$, $z_{vk}$, $z_k$= 0, $z_{qk}$= 0, $z_{bk}$= 0, consider $N_k[0]$ as depot.

3: Set $z_k = z_{ck} + z_{vk}$

4: **for** nodes in positions $i$=1 to $|N_k|$ do

5: Determine the service starting time ($T_{N_k[i]}$).

6: **if** $T_{N_k[i]} \leq l_{N_k[i]}$ and $T_{N_k[i]} \geq e_{N_k[i]}$ **then**

7: Determine $\max\{T_{N_k[i]}, a_{N_k[i]}\}$

8: Obtain $sl - \max\{T_{N_k[i]}, a_{N_k[i]}\}$

9: Set $q_{N_k[i]} = \frac{sl - \max\{T_{N_k[i]}, a_{N_k[i]}\}}{sl}$

10: **if** $q_{N_k[i]} \geq ql_{N_k[i]}$ **then**

11: Determine $(1 - q_{N_k[i]})\, p\, d_{N_k[i]}$

12: Set $z_{qk} = z_{qk} + (1 - q_{N_k[i]})\, p\, d_{N_k[i]}$

13: **if** $T_{N_k[i]} - b_{N_k[i]} > 0$ **then**

14: $z_{bk} = z_{bk} + c_b(T_{N_k[i]} - b_{N_k[i]})$

15: **end if**

16: Set $z_k = z_k + z_{qk} + z_{bk}$

17: **else**

18: “Route is infeasible“, and go to 24

19: **end if**

20: **else**

21: “Route is infeasible“, and go to 24

22: **end if**

23: **end for**

24: **End**

---

In the pseudocode above, it should be noted that lines 1-3 initialize the algorithm with the starting values of cost parameters and sets. Lines 4 and 5 determine the service starting time for the nodes in route *k*. The time window constraints are checked in line 6. If the time window constraint is satisfied for the route under consideration, then the algorithm proceeds to the next step for checking the feasibility of the route for the minimum preferred quality level imposed by the customer at that node in lines 7-10. Otherwise, the route is infeasible, and the algorithm is terminated at line 24. If the route is found to be feasible under both time window and quality constraints, then it is a feasible route and a feasible solution for TSPSPD-STW-P as well. The loss-of-quality cost and penalty cost for late arrival are calculated in lines 11-16. The total cost of route *k* is reported in line 17 by summing the four cost elements: traveling and vehicle costs that were already obtained by the solution of TSPSPD, and the loss-of-quality and late arrival penalty costs found here in this algorithm.

The pseudocode of the overall solution approach for obtaining a feasible solution for the main problem VRPSPD-STW-P from *m* subtours of a giant tour is given in the following algorithm.

---

**Algorithm: The overall solution approach for obtaining a feasible solution for the main problem VRPSPD-STW-P**

---

1:**Begin**

2: Apply the clustering algorithm to a generated giant tour.

3: **for** routes $k$=1 to $m$ **do**

4: Set $z_{ck}$= 0, $z_{vk}$= 0, $z_{qk}$= 0, $z_{bk}$= 0, $z_k$= 0, $z$ = 0, and $N_k$[0] as the single depot.

5: Perform one of the routing heuristic algorithms for route *k*.

6: **for** nodes in positions $i$=1 to | $N_k$ | **do**

7: Implement the algorithm for the TSPSPD for route *k*.

8: **if** route is a feasible solution for the TSPSPD **then**

9: Set $z_k = z_{ck} + z_{vk}$

10: Implement the algorithm for the TSPSPD-STW-P for route *k*.

11: **if** route is a feasible solution for the TSPSPD-STW-P **then**

12: Set $z_k = z_k + z_{qk} + z_{bk}$

13: Set $z = z + z_k$

14: **else**

15: "Route is infeasible".

16: " The generated solution with *m* routes is not a feasible solution for the main problem, VRPSPD-STW-P".

17: go to 27

18: **end if**

19: **else**

20: "Route is infeasible".

21: "The generated solution with *m* routes is not a feasible solution for the main problem, VRPSPD-STW-P".

22: go to 27

23: **end if**

24: **end for**

25: **end for**

26: "The generated solution with *m* routes is a feasible solution for VRPSPD-STW-P"

27: **End**

---

Lines 1 and 2 perform the clustering algorithm for a generated giant tour. Line 3 enumerates *m* routes in the clustered giant tour. Line 4 initializes the algorithm with some starting parameters and sets. The routing algorithm is performed in line 5. The implementation of the algorithms starts in line 6 and ends in line 27. The feasibility of each route of the giant tour is checked independently. In lines 7-9, each route *k* is checked to see whether it is a feasible solution for the TSPSPD. If it is found to be feasible, then the route $k$ is checked to see whether it is a feasible solution for the TSPSPD-STW-P in lines 10-13. If any route is infeasible, the generated solution, which includes $m$ routes of the giant tour, is infeasible, and the overall algorithm is terminated; otherwise, the generated solution is a feasible solution for the main problem VRPSPD-STW-P.

In the next sections, we first discuss the approach for obtaining the pool of initial feasible solutions. Then we generate several improved solutions for the main problem VRPSPD-STW-P by using the solution approach discussed above in section 4.2.3 and the genetic algorithm discussed in section 4.4 below.

**4.3. Phase 1: Initial feasible solutions pool**

As mentioned in the solution approach section, all nodes of the problem are considered a giant route, which is then partitioned tour into $m$ routes (clusters) such that each route is a candidate solution (feasible/infeasible) for the TSPSPD-STW-P. We check the feasibility of each route independently of others. The solution results of all routes (clusters), if each route is feasible, turn out to be the VRPSPD-STW-P solution, and the summation of objective functions of routes is VRPSPD-STW-P objective function value.

To summarize, for checking the feasibility of the solution, each generated route is tested for TSPSPD solution separately; if all routes are found to be feasible, they are checked then for soft time window and quality constraints by the TSPSPD-STW-P algorithm. This feasibility check procedure is employed for all $m$ routes separately. If all routes are feasible, then we obtain a feasible solution for the main problem, VRPSPD-STW-P. Therefore, a feasible solution of VRPSPD-STW-P is composed of $m$ feasible solutions such that each of them is a feasible solution for a TSPSPD-STW-P. The feasible solutions, thus obtained, are kept in the pool of feasible solutions for the main problem VRPSPD-STW-P.

Therefore, the initial population is generated based on random clustering of the giant tour and sequencing of customer nodes in each route using one of the routing heuristics as discussed in the previous sections. The pseudocode for Phase 1 of the solution approach is given in the following algorithm. Letting the total number of generated solutions be $g$, then $g$ many solutions are generated and the total number of feasible solutions required in Phase 1 be denoted by *sample*, then we need to generate initial feasible solutions as many as the *sample*.

---

**Algorithm: Phase 1: Initial feasible solutions pool**

---

1: **Begin**

2: Set $g$, *sample*, *sa*= 1, *feasible*= 1.

3: **while** initial generation counter $sa \leq g$ or $feasible \leq sample$ **do**

4: Generate a routing sequence of all nodes randomly, and call it a giant tour.

5: Perform the clustering algorithm for the generated giant tour.

6: **for** routes $k$=1 to $m$ **do**

7: Set $z_{ck}$= 0, $z_{vk}$= 0, $z_{qk}$= 0, $z_{bk}$= 0, $z_k$= 0, $z$ = 0, consider $N_k[0]$ as depot.

8: Perform one of the routing heuristic algorithms for route $k$.

9: **for** nodes in positions $i$=1 to $| N_k |$ **do**

10: Implement the algorithm for TSPSPD of route $k$.

11: **if** the route is feasible **then**

12: Set $z_k = z_{ck} + z_{vk}$

13: Implement the algorithm for TSPSPD-STW-P of route $k$.

14: **if** the route is feasible **then**

15: Set $z_k = z_k + z_{qk} + z_{bk}$

16: Set $z = z + z_k$

17: **else**

18: "Route is infeasible".

19: " The generated solution with $m$ routes is not feasible solution for VRPSPD-STW-P".

20: go to 36.

21: **end if**

22: **else**

23: “Route is infeasible“.

24: “The generated solution with *m* routes is not feasible solution for VRPSPD-STW-P”.

25: go to 36.

26: **end if**

27: **end for**

28: **end for**

29: “The generated solution with *m* routes is a feasible solution for the main problem, VRPSPD-STW-P”.

30: Add this solution to the initial solutions pool.

31: *feasible*= *feasible* + 1.

32: **if** *feasible* = *sample* **then**

33: “ Required number of initial feasible solutions are in the pool”

34: go to 39.

35: **end if**

36: *sa*= *sa* +1

37: **end while**

38: “No feasible solutions for the required *sample*”

39: **End**

---

Lines 1 and 2 initialize the algorithm to obtain initial feasible solutions with some starting parameters and sets. *feasible* is the feasible solution counter, *sample* is the total number of initial feasible solutions requested, *sa* is the initial phase generation counter, and the maximum limit for *sa* is *g*. Line 3 indicates two stopping conditions for the algorithm and checks the number of generations and the number of feasible solutions against the required numbers of those. Generating a giant tour and clustering it as *m* routes are performed in lines 4 and 5. Line 6 enumerates the *m* routes in the clustered giant tour for checking the feasibility of each independently. Line 7 initializes the algorithm with the cost parameters and the nodes set. The routing algorithm is performed in line 8 for the route under consideration. The overall algorithm proceeds from line 9-28, as it was discussed in the solution approach for obtaining a feasible solution of VRPSPD-STW-P. In lines 9-12, the solution for route $k$ is obtained for the problem TSPSPD, and if it is found to be feasible, the solution for route $k$ is obtained for TSPSPD-STW-P in lines 13-16.

If the generated solution is found to be feasible at the end, it is added to the initial solutions pool, and the algorithm checks the stopping conditions. If the generated solution is infeasible, it is discarded from the process, and again the algorithm checks the stopping conditions. The algorithm is terminated when $sa=g+1$ or $feasible=sample+1$. The maximum number of generations, $g$, is one of the stopping conditions for the algorithm. The other stopping criterion is the number of initial feasible solutions required, *sample*, for forming the initial pool of GA in Phase 2. When the algorithm cannot obtain the required number of feasible solutions among the generations as many as the specified number, it is reported “No feasible solutions for the required sample”. When the required number of initial feasible solutions in the pool reaches the *sample*, it is reported “the required number of initial feasible solutions is in the pool” and the algorithm is terminated. Then the solution methodology proceeds with Phase 2 where we develop a GA algorithm to improve the feasible solutions in the initial pool.

### 4.4. Phase 2: Solution improvement-Genetic Algorithm

The initial solutions obtained in Phase 1 of the methodology above form a pool of initial feasible solutions that can be improved using different approaches. In Phase 2, we develop a genetic algorithm (GA) for improving the initial solutions. After generating new solutions in chromosome forms in the pool of the GA, we employ the same algorithm as we do in Phase 1 in order to check whether the solution generated is a feasible one. The best results are compared with each other for obtaining the best solutions in terms of cost.

### 4.4.1. Chromosome definition

In general, GA searches for better solutions by starting with the randomly generated set of initial feasible solutions encoded to *chromosomes*. The total number of feasible chromosomes in the genetic pool is the required number of initial feasible solutions as denoted by *sample*. In this algorithm, each chromosome has a length that is equal to the number of customers, and each gene in the chromosome represents a vehicle. A newly generated chromosome in each generation of the GA is an encoded set that represents a new solution for VRPSPD-STW-P when decoded.

We define that *Nodes* is a sorted set (array) of customer nodes in the ascending order of their numbers, where the position of node 1 is *Nodes* [0], node 2 position is *Nodes* [1], and similarly the other nodes positioned, and end with node *n* position in *Nodes* [*n*-1]. Each node in *Nodes* set in position *i* is served by its corresponding vehicle in the chromosome set in position *i*. Therefore, in our encoding, every chromosome is a string of vehicle numbers, in which each gene contains a vehicle number and the gene position indicates its customer in *Nodes* set.

To decode the chromosome set, we need to define *NODES* set and the *vehicl*e set. *NODES* set includes the clustered routes, starting the sequence with the nodes of the first vehicle (vehicle 0), and then the second vehicle (vehicle 1), and goes on with the other vehicles. This *NODES* set alone is not sufficient to decode unless the *vehicl*e set is defined with the vehicle numbers it contains. On the other hand, the *vehicl*e set includes the vehicle numbers in sequence starting with 0, and going on with 1, 2, etc. Each vehicle number is repeated as many as the number of nodes it is serving.

The *vehicle* set contains the vehicles that are serving their corresponding nodes in the *NODES* set. For example, the following *NODES* set and *vehicle* set are encoded as in the following *Nodes* and *chromosome* sets.

$$NODES = \{5\ 8\ 2\ 6\ 7\ 10\ 1\ 3\ 4\ 9\}$$

$$vehicle\ = \{0\ 0\ 1\ 1\ 1\ \ 1\ \ 2\ 2\ 2\ 2\}$$

Its corresponding chromosome set is:

$$\begin{aligned} \textit{Nodes} &= \{1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\} \\ \textit{chromosome} &= \{2\ 1\ 2\ 2\ 0\ 1\ 1\ 0\ 2\ \ 1\} \end{aligned}$$

So, it is seen that three vehicles are to serve ten customer nodes as follows: node 1 by vehicle 2, node 2 by vehicle 1, node 3 by vehicle 2, and so on.

### 4.4.2. Genetic operators

As the evolution process, two genetic operators are derived to generate new chromosomes in each generation. This evolutionary process contains operations of crossover and mutation. The crossover and mutation operations are carried out by picking chromosomes from the pool based on their objective function costs. Some candidates are then mated to produce offspring by crossover, and some go through a mutating process.

Crossover is implemented by selecting one, two or multiple random points on the chromosome where the parents' gene exchange occurs. The mutation applies the changes randomly to one or more genes to produce a new offspring, thus generating new adaptive solutions to avoid local optima. Normally, the mutation operation takes place after the crossover; that is a matter of preference.

The fitness function in the genetic algorithm is used to evaluate the performance of a chromosome. If the chromosome is feasible, we can evaluate its fitness by the cost obtained from the objective function. Hence, we define the fitness function of the GA by the objective function value. Fitness value for a solution with *m* routes is:

$$FV^* = \{ \sum_{k \in K} z_k \},\ k=\{0,1,\ldots,m\text{-}1\}$$

where $z_k$ is the total cost of route *k*.

In the first step for implementing genetic operators in each generation, in order to avoid losing the highest fitness valued chromosomes, elitism has been implemented. Elitism stores some highest-scoring chromosomes of the current generation for generating the next generation without allowing that chromosome to be crossed over and mutated. A certain number of parents with the lowest costs is chosen and left in the pool with no change. After that, a certain number of parents with the lowest cost are picked for performing the well-known two-point crossover and mutation operators to generate new chromosomes.

Crossover is applied by exchanging the genes of a pair of chromosomes. The mutation method is based on randomly picking some genes in the chromosome and alter gene values from their initial values. The order-based mutation is also adopted to produce heterogeneous chromosomes in the pool in order to avoid the early convergence of the algorithm. In the first step of using GA operators, the crossover is performed, and then mutation is made on the child chromosomes.

In GA, mutation operators are mostly used to provide exploration, whereas crossover operators are mostly used to provide exploitation and converge on a good solution. Consequently, while crossover tries to converge to a specific point in solution space, the mutation does its best to avoid convergence and explore more areas.

Using the crossover operator alone to produce an offspring makes the GA stuck in the local optima. The mutation operator is designed to help the search escape from local optima; that is, the mutation operator is used to generate new offspring different from the parents, and thus encouraging diversity in the population. We prefer to explore much more at the beginning of the search process to ensure population coverage and diversity. On the other hand, we prefer more exploitations at the search process to ensure the convergence of the population to the global optimum.

During evolution, the crossover and mutation operations occur according to the specified probabilities, named as *crossover probability* ($p_c$) and *mutation probability* ($p_m$). The crossover probability gives the fraction of chromosomes actually crossed in one generation, i.e., the probability indicates a ratio of how many chromosomes in one population are picked for mating by the crossover operation. The probability of mutation determines the probability that a gene is mutated in one generation. Typically mutation happens with a very low probability, such as 0.001 to 0.05.

The probability of 100% crossover means that the newly generated population is made by crossover. If the probability is 0%, then the completely new generation of chromosomes is exactly copied from the older population. On the other hand, the probability of 10% mutation means that 10 randomly picked genes of all 100 genes in a population are changed.

Consider an example of crossover, for instance, with 10 customers to be visited by three vehicles. Figure 4.3 shows crossover and mutation operators for this example. In this example, two points, say 5 and 7, are selected for crossover cut points, and the genes in the left-hand side of point 5 and the right-hand side of point 7 of opposite parents are exchanged. The mutation operation can be seen in parents 1 and 2, the $5^{th}$ gene of parent 1, and the $10^{th}$ gene of parent 2, which is shown in bold in Figure 4.3. The decoding of child chromosomes is shown in Figure 4.4.

Assume $p_c = 0.5$ and $p_m = 0.05$; hence we expect to implement a crossover operator for 50% of chromosomes, and approximately 5% of genes are inverted by mutation operator in the population. Suppose the population size is 20. The whole population is sorted according to the chromosome costs.

For creating a new population, the elitist selection strategy is used, and we keep 10 parents with the highest fitness value (lowest cost) with no change in the pool. These 10 parents with minimum cost are paired up for implementing crossover, and then 10 genes on the child chromosomes, which are generated by crossover, are mutated randomly for generating a new population. Therefore, there are 200 genes in the population, and 10 genes out of 200 are mutated.

| | |
|---|---|
| Parent 1 | 1 0 2 0 \| 0 2 1 \| 1 1 2 |
| Parent 2 | 2 0 1 1 \| 1 2 1 \| 1 0 0 |
| Child 1 | 2 0 1 1 \| **2** 2 1 \| 1 0 0 |
| Child 2 | 1 0 2 0 \| 1 2 1 \| 1 1 **1** |

Figure 4.3. An example for crossover and mutation

| | |
|---|---|
| *Nodes* 1 | 1 \| 2 \| 3 \| 4 \| 5 \| 6 \| 7 \| 8 \| 9 \| 10 |
| *NODES* for child 1 | 2 9 10 \| 3 4 7 8 \| 1 5 6 |
| *Nodes* 2 | 1 \| 2 \| 3 \| 4 \| 5 \| 6 \| 7 \| 8 \| 9 \| 10 |
| *NODES* for child 2 | 2 4 \| 1 5 7 8 9 10 \| 3 6 |

Figure 4.4. An example for decoding of child chromosomes

Results of crossover and mutation on two chromosomes are as follows:

Parent 1:

$N_0$ =$Route_0$=[2 4 5]

$N_1$ =$Route_1$=[ 1 7 8 9]

$N_2$ =$Route_2$=[ 3 6 10]

Parent 2:

$N_0$ =$Route_0$=[ 2 9 10 ]

$N_1$ =$Route_1$=[ 3 4 5 7 8 ]

$N_2$ =$Route_2$=[ 1 6 ]

Child 1:

$N_0$ =$Route_0$=[ 2 9 10 ]

$N_1$=$Route_1$=[ 3 4 7 8 ]

$N_2$ =$Route_2$=[ 1 5 6 ]

Child 2:

$N_0$ =$Route_0$=[ 2 4 ]

$N_1$ =$Route_1$=[ 1 5 7 8 9 10 ]

$N_2$ =$Route_2$=[ 3 6 ]

After decoding the chromosome and determining *Route*$_k$ sets, the algorithms for checking the chromosome's feasibility are implemented, starting with the routing of nodes in each route.

The objective function value of each chromosome (solution) is obtained using the solution approach discussed in section 4.2.3 and used to obtain initial solutions. Among the feasible chromosomes, the one with the minimum objective function value is saved as the solution of the current generation. After that, the generation counter (*generation)* is increased by one unit. If the generation counter is equal to *gen*+1, the algorithm terminates; otherwise, a new pool of chromosomes is generated through a guided evolution.

The pseudocode and detailed steps of the GA to solve the main problem (with the maximum number of genetic generation *gen,* generation counter *generation, generation*={1,…, *gen*}, and chromosome counter *sa, sa*={1,…, *sample*}) are illustrated in the following algorithm.

---

**Algorithm: Phase 2: Solution improvement-Genetic algorithm**

---

1: **Begin**

2: Set *gen*, *sample,* $p_c$, $p_m$ and *generation* =1, *sa*= 1, $z_{min}$ = minimum objective function value in the pool.

3: **while** genetic generation counter *generation* ≤ *gen* **do**

4: Sort all feasible solutions in the pool in the ascending order of their objective function values.

5: Generate a chromosomes pool of solutions (encoding).

6: Keep a certain number of chromosomes with the minimum objective function values without any change in the pool.

7: Implement two-point crossover operation for a certain number of chromosomes which have the minimum objective values in the pool.

8: Implement mutation operation for a certain number of genes that are randomly picked in child chromosomes generated in step 7 and change their values.

9: **while** chromosomes *sa* =1 to *sample* **do**

10: Decode the chromosome numbered *sa*.

11: Set $N_k$ and $|N_k|$.

12: **for** routes $k$=1 to $m$ **do**

13: Set $z_{ck}$= 0, $z_{vk}$= 0, $z_{qk}$= 0, $z_{bk}$= 0, $z_k$= 0, $z$ = 0, consider $N_k[0]$ as depot.

14: Perform one of the routing heuristic algorithms for route $k$.

15: **for** nodes in positions $i$=1 to $|N_k|$ **do**

16: Perform solution approach algorithm for the TSPSPD problem of route $k$.

17: **if** the route is feasible **then**

18: Set $z_k = z_{ck} + z_{vk}$

19: Perform the solution approach algorithm for the TSPSPD-STW-P of route $k$.

20: **if** the route is feasible **then**

21: Set $z_k = z_k + z_{qk} + z_{bk}$

22: Set $z = z + z_k$

23: **else**

24: "Route is infeasible".

25: "The generated solution with $m$ routes is infeasible solution for VRPSPD-STW-P, and then chromosome is infeasible".

26: go to 39.

27: **end if**

28: **else**

29: "Route is infeasible".

30: "The generated solution with $m$ routes is infeasible for VRPSPD-STW-P and then chromosome is infeasible".

31: go to 39.

32: **end if**

```
33:                  end for
34:            end for
35:            "The generated solution with m routes is feasible".
36:            if z < z_min then
37:                  Set z_min = z
38:            end if
39:            sa= sa +1
40:      end while
41:      generation = generation +1
42: end while
43: Set FV** = z_min as the minimum objective function value.
44: End
```

Line 1 and 2 initialize the algorithm for some starting parameters and counters. The total number of chromosomes in the pool is the population size, which equals the number of feasible solutions (*sample*). *sample* denotes the total number of initial feasible solutions required in the pool, *generation* is the generation counter in the GA with *gen* as the maximum limit, while *sa* is the chromosome counter in the pool.

Line 3 indicates a stopping condition for the GA algorithm and tallies the generations. In lines 4 and 5 the gene pool is formed, solutions are expressed by chromosomes, and the encoding of solutions is performed. Lines 6-8 represent the evolutionary process for generating new chromosomes in the pool, which contains both crossover and mutation operations for forming a new gene pool. Checking the feasibility of the generated chromosomes starts with line 9; this line tallies the newly generated chromosomes as solutions of the problem. In lines 10 and 11, chromosomes are decoded, and the solution approach algorithm is started by the parameters and sets.

The decoded chromosome gives a clustered tour directly. Line 12 tallies $m$ routes in the clustered giant tour for checking the feasibility of each route independently. Line 13 initializes the algorithm with the starting values of cost parameters and sets. The routing algorithm is performed in line 14. The overall algorithm starts from line 12-35 as it is discussed in the solution approach for obtaining a feasible solution for the main problem, VRPSPD-STW-P. In lines 16-18, the TSPSPD solution for route $k$ is obtained, and if it is feasible, then the TSPSPD-STW-P solution for route $k$ is found in lines 19-22.

When a generated solution is feasible, its objective function value is compared with the minimum solution found so far until the current generation. If the new value is less, its value replaces the previous best value. After checking the feasibility of all chromosomes in the gene pool, the algorithm checks the stopping criterion. The maximum number of genetic generation (*gen*) is the stopping condition for the problem, and the algorithm is terminated when *generation*=*gen*+1. The solution with the minimum objective value of all generations is the best solution obtained by our solution methodology, and $FV^{**}= z_{min}$.

In the following chapter, we present and analyze the results of our computational study. We generate several problem instances, differing in terms of vehicle capacities, distances between nodes, time windows of customers, and quality levels required by customers. Thus, we solve the problem instances thus generated for VRPSPD-STW-P by the exact solution approach, MILP using CPLEX and the proposed methodology.

# CHAPTER 5

# COMPUTATIONAL STUDY

We conduct computational experiments with several problem instances to test the performance of our solution methodology. Firstly, the designed instances with their parameter settings are described. Then, computational results and the sensitivity analyses are reported in the following sections.

## 5.1. Problem instances

The problem instances in the literature are mostly for VRP and some of its extensions like VRPTW, which are not exactly suitable for the problem VRPSPD-STW-P we address. In this problem, we also need pickup and delivery requirements, time window limitations, product quality requirements of the demand nodes, route length limitation, and vehicle capacity.

As a result, there is not any benchmark study in the literature that exactly matches the VRPSPD-STW-P addressed in this study. For evaluating our proposed approach, several instances are generated randomly for the problem VRPSPD-STW-P.

Hence, we generate the necessary parameters of the problem randomly in the specified ranges to obtain several problem instances. All problem instances are generated in C# language using Microsoft Visual Studio.12.0 on a personal computer with an Intel Core i7 CPU and 12 GB RAM.

For generating the parameters, the uniform distribution is used in different ranges. To bring the problem instances as close as possible to real life, we consider both small-to-medium size retailers and large retailers in generating the problem instances. Therefore, in defining the ranges for small-to-medium size retailers, we use tighter ranges for travel distances and times on the network, time window ranges, pickup and delivery demands, and other parameters accordingly; on the other hand, we use wider ranges for these parameters for large retailers.

After generating the parameters, they are checked for their validity in describing a problem setting. For this validity checking, each generated instance is solved by any one of the proposed solution approaches in only a limited number of iterations; if the problem instance tested for validity leads to a reasonable solution, we include it in the computational study as a problem instance.

**Large-retailer instances:** Consider an instance with $n$=50 demand points designed for large size retailers. The uniform distributions for the parameters are set as follows: distance between any two demand points ($w_{ij}$) range of [10, 200], travel time between any two demand points ($t_{ij}$) range of [20,50], delivery demand for a demand point ($d_i$) range of [100,500] and pickup demand for a demand point ($z_i$) range of [20,100]. Considering these ranges, we define capacity ($Q$) and travel distance ($L$) limitations in the ranges within [3000,3500] and [2000,2500] respectively. In the next step we generate ranges for time windows: start of time window for arriving at customer $i$ ($a_i$) range of [150,250], end of time window for arriving at customer $i$ ($b_i$) range of [250,600], earliest acceptable time for arrival at customer $i$ ($e_i$) range of [10,150], and latest acceptable time for arrival at customer $i$ ($l_i$) range of [600,700].

**Small-to-medium-retailer instances:** Consider an instance with $n$=20 demand points designed for small-to-medium size retailers. The uniform distributions for the parameters are set as follows: distance between any two demand points ($w_{ij}$) range of [1,10], travel time between any two demand points ($t_{ij}$) range of [1,10], delivery demand for a demand point ($d_i$) range of [15,25] and pickup demand for a demand point ($z_i$) range of [1,5]. Considering these ranges, we define capacity ($Q$) and travel distance ($L$) limitations in the ranges within [150,200] and [100,150] respectively. In the next step we generate ranges for time windows: start of time window for arriving at customer $i$ ($a_i$) range of [25,40], end of time window for arriving at customer $i$ ($b_i$) range of [50,70], earliest acceptable time for arrival at customer $i$ ($e_i$) range of [1,20], and latest acceptable time for arrival at customer $i$ ($l_i$) range of [100,210].

As for the price and cost parameters, we choose to use the following scenarios for all problem instances: high price-high cost, low price-low cost, high price-low cost, and low price-high cost.

For the lowest acceptable quality level by customer $i$ ($q_i$), we consider a fixed value, $q_i$=%30, for all customers in all problem instances. The effects of changes in the lowest acceptable quality, $q_i$, are discussed in sensitivity analysis section where it is changed from %10 to %90.

### 5.2. Exact method

To compare the performance of our solution methodology against the optimal solutions, we use the solver CPLEX, version 12.7.0.0 of IBM ILOG CPLEX Optimization Studio. The MILP model of the main problem VRPSPD-STW-P is implemented using CPLEX, and the problem instances with sizes of 9 and 11 demand nodes (customers) are solved. Attaining optimal solution for small size problems is possible by CPLEX for the MILP model. However, it is observed that solution times for the MILP model increase exponentially for the problems with the number of nodes higher than 10.

Here we present and discuss the optimal solutions for the test problems with the number of nodes being $n$=9 and $n$=11 (Appendix A).

Tables 5.1, 5.2, 5.3, and 5.4 show computing times and objective values for different numbers of vehicles allowed ($m$) followed by the solution details for the two test problems with the number of nodes 9 and 11, respectively. Figures 5.1, 5.2, 5.3, and 5.4 present the quality levels vs starting service times for the four routes in the test problem with $n$=9 nodes.

Table 5.1. Optimal solution for the problem with $n$=9 nodes

| $m$ | Computing time (s) | Objective value | $V_a$ | $V_r$ |
|---|---|---|---|---|
| 8 | 4.64 | 387.04 | 2 | 2 |
| 7 | 3.63 | 387.04 | 2 | 2 |
| 6 | 2.96 | 387.04 | 2 | 2 |
| 5 | 2.08 | 387.04 | 2 | 2 |
| 4 | 1.47 | 387.04 | 2 | 2 |
| 3 | 1.17 | 418.39 | 2 | 1 |
| 2 | - | infeasible | - | - |

Optimal Routes with $V_a$ =2 owned vehicles and $V_r$ =2 rented vehicles:

Route 1 (vehicle 1): 0--2--1--8--0
Route 2 (vehicle 2): 0--3--6--5--0
Route 3 (vehicle 3): 0--4--9--0
Route 4 (vehicle 4): 0--7--0

Table 5.2. Solution details for the problem with $n$=9 nodes

| Nodes | Route | vehicle | $T_{0k}$ | $T_{ik}$ | $(T_{ik} - b_i)^+$ | $q_i$ | $z_i$ | $d_i$ | $Y_{ijk}$ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0--2 | 1 | 2 | 0 | 0 | 1 | 0 | 0 | 87 |
| 2 | 2--1 | 1 | 0 | 5 | 0 | 0.9 | 7 | 30 | 64 |
| 1 | 1--8 | 1 | 0 | 9 | 0 | 0.82 | 3 | 20 | 47 |
| 8 | 8--0 | 1 | 0 | 12 | 0 | 0.76 | 6 | 37 | 16 |
| 0 | 0--3 | 2 | 1 | 0 | 0 | 1 | 0 | 0 | 75 |
| 3 | 3--6 | 2 | 0 | 6 | 0 | 0.88 | 8 | 20 | 63 |
| 6 | 6--5 | 2 | 0 | 14 | 0 | 0.72 | 20 | 40 | 43 |
| 5 | 5--0 | 2 | 0 | 17 | 5 | 0.66 | 6 | 15 | 34 |
| 0 | 0--4 | 3 | 0 | 0 | 0 | 1 | 0 | 0 | 65 |
| 4 | 4--9 | 3 | 0 | 7 | 0 | 0.86 | 2 | 25 | 42 |
| 9 | 9--0 | 3 | 0 | 18 | 0 | 0.64 | 21 | 40 | 23 |
| 0 | 0--7 | 4 | 2 | 0 | 0 | 1 | 0 | 0 | 20 |
| 7 | 7--0 | 4 | 0 | 9 | 0 | 0.82 | 9 | 20 | 9 |

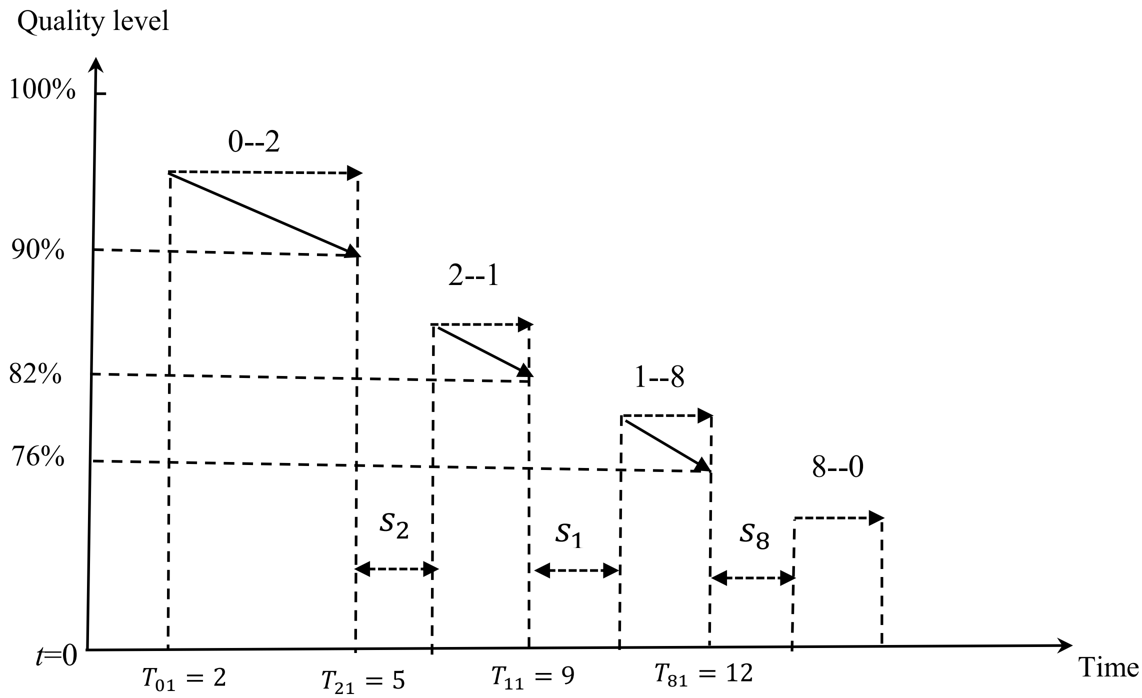



Figure 5.1. Quality levels vs starting service times for route 1 for the problem with $n$=9 nodes

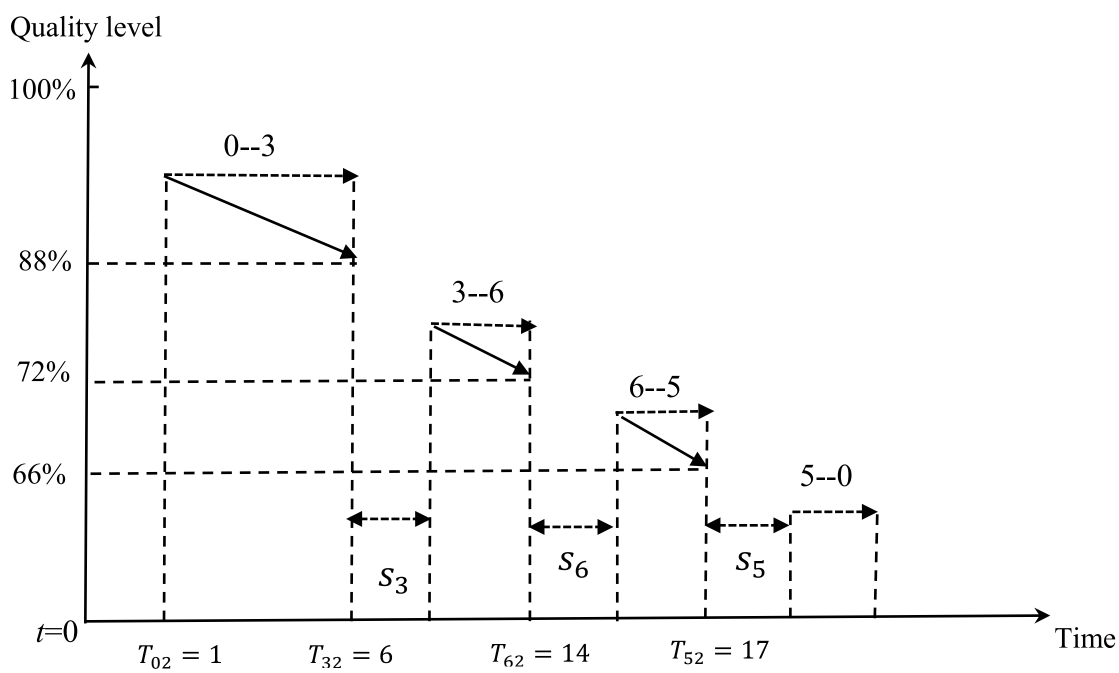



Figure 5.2. Quality levels vs starting service times for route 2 for the problem with $n$=9nodes

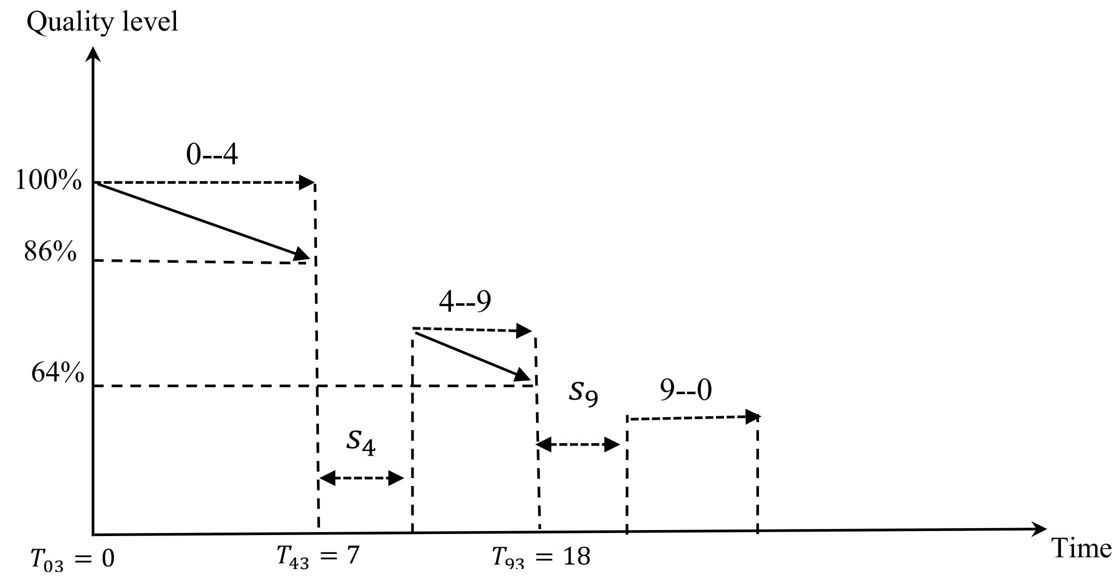



Figure 5.3. Quality levels vs starting service times for route 3 for the problem with $n$=9 nodes

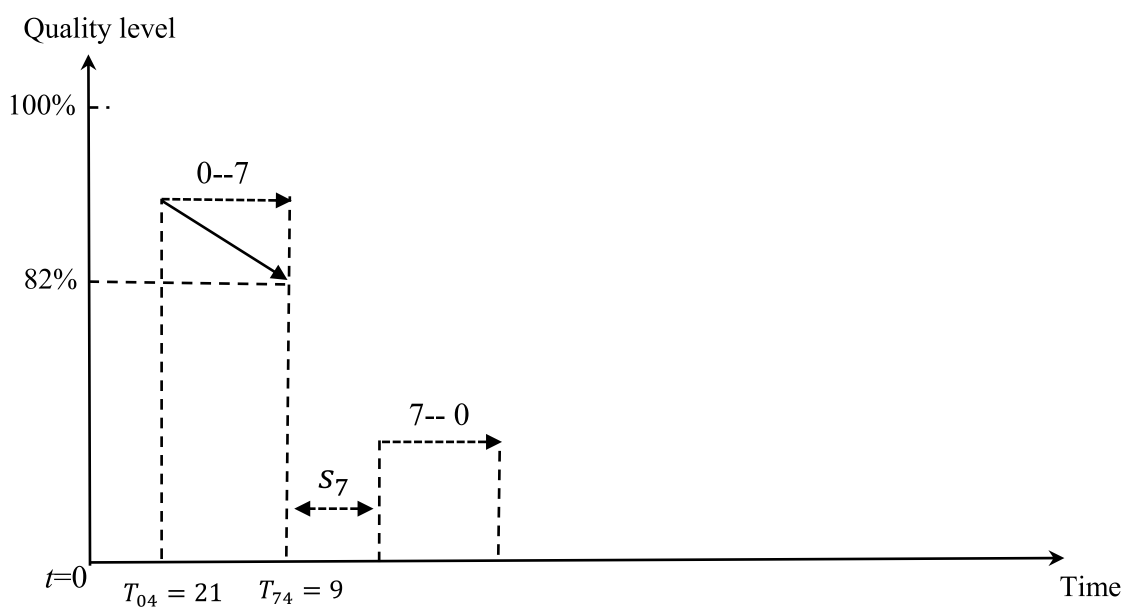



Figure 5.4. Quality levels vs starting service times for route 4 for the problem with $n$=9 nodes

Table 5.3. Optimal solution for the problem with $n$=11 nodes

| $m$ | Computing time (s) | Objective value | $V_a$ | $V_r$ |
|---|---|---|---|---|
| 9 | 33.33 | 495.08 | 2 | 3 |
| 8 | 27.62 | 495.08 | 2 | 3 |
| 7 | 5.09 | 495.08 | 2 | 3 |
| 6 | 4.06 | 495.08 | 2 | 3 |
| 5 | 2.57 | 495.08 | 2 | 3 |
| 4 | 2.95 | 588.78 | 2 | 2 |
| 3 | - | infeasible | - | - |

Table 5.4. Solution details for the problem with $n$=11 nodes

| Node | Route | vehicle | $T_{0k}$ | $T_{ik}$ | $(T_{ik}-b_i)^+$ | $q_i$ | $z_i$ | $d_i$ | $Y_{ijk}$ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0--2 | 1 | 2 | 0 | 0 | 1 | 0 | 0 | 87 |
| 2 | 2--1 | 1 | 0 | 5 | 0 | 0.9 | 7 | 30 | 64 |
| 1 | 1--8 | 1 | 0 | 9 | 0 | 0.82 | 3 | 20 | 47 |
| 8 | 8--0 | 1 | 0 | 12 | 0 | 0.76 | 6 | 37 | 16 |
| 0 | 0--3 | 2 | 1 | 0 | 0 | 1 | 0 | 0 | 75 |
| 3 | 3--6 | 2 | 0 | 6 | 0 | 0.88 | 8 | 20 | 63 |
| 6 | 6--5 | 2 | 0 | 14 | 0 | 0.72 | 20 | 40 | 43 |
| 5 | 5--0 | 2 | 0 | 17 | 0 | 0.66 | 6 | 15 | 34 |
| 0 | 0--4 | 3 | 0 | 0 | 0 | 1 | 0 | 0 | 65 |
| 4 | 4--9 | 3 | 0 | 7 | 0 | 0.86 | 2 | 25 | 42 |
| 9 | 9--0 | 3 | 0 | 18 | 0 | 0.64 | 21 | 40 | 23 |
| 0 | 0--7 | 4 | 1 | 0 | 0 | 1 | 0 | 0 | 71 |
| 7 | 7--10 | 4 | 0 | 9 | 0 | 0.82 | 9 | 20 | 60 |
| 10 | 10--0 | 4 | 0 | 18 | 0 | 0.64 | 8 | 51 | 17 |
| 0 | 0--11 | 5 | 7 | 0 | 0 | 1 | 0 | 0 | 24 |
| 11 | 11--0 | 5 | 0 | 9 | 0 | 0.82 | 2 | 24 | 2 |

The vehicle routes of the two test problems are depicted in Figures 5.5 and 5.6. Each route in the solutions of test problems corresponds to a solution of TSPSPD-STW-P with the nodes covered on that route. The solution of VRPSPD-STW-P is then composed of *m* many solutions of mTSPSPD-STW-P for that problem instance.

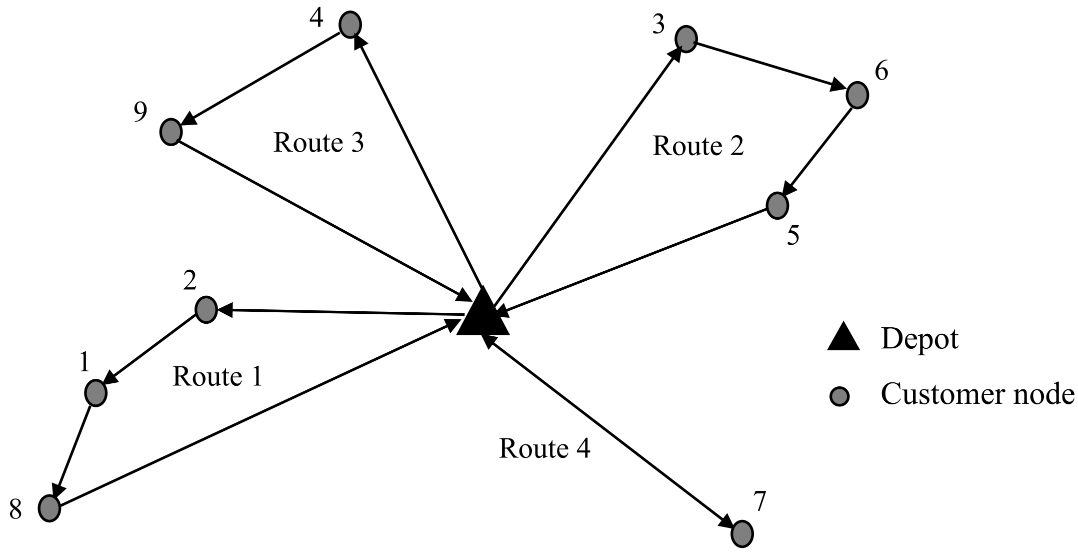



Figure 5.5. Vehicle routes for the problem with $n$=9 nodes

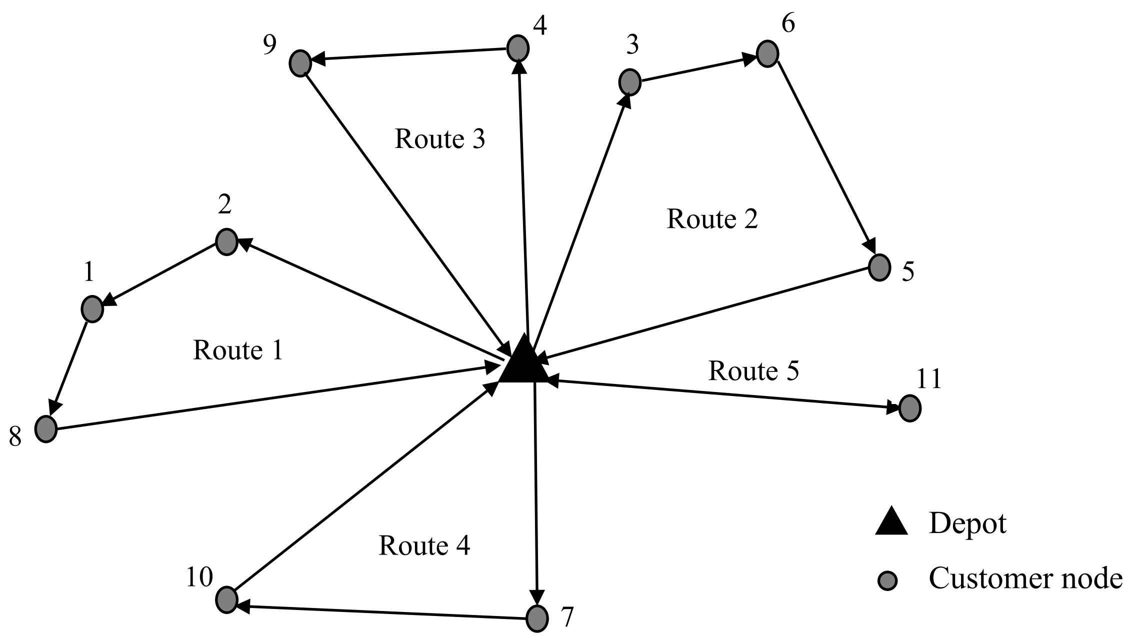



Figure 5.6. Vehicle routes for the problem with $n$=11 nodes

By considering the alternative feasible values for *m*, CPLEX provides either an optimal solution or a feasible solution as an upper bound for the optimal solution. However, we should note that we aim to obtain the minimum cost solution to satisfy the demands and quality requirements of the customers by using the least number of vehicles within a reasonable computing time.

The computational results show that optimal or near-optimal solutions for large-size problems can only be obtained using approximate methods rather than the exact method; for example, the problem with 15 nodes could not be solved by CPLEX in reasonable computing time. It is well known from the literature that the exact algorithms are limited for solving instances of large size even when the problem is the basic VRP ignoring the additional characteristics of our problem as simultaneous pick-ups and deliveries, time windows, and quality levels required upon delivery at the customer site.

### 5.3. Proposed solution approaches

In our proposed solution methodology, we use any one of the three different clustering and routing heuristics in Phase 1 and then use the routing heuristics plus GA in Phase 2. Hence we end up with three different solution approaches only differentiated by the type of clustering and routing heuristic as follows:

- Random clustering and routing (RCR) & GA
- Independent clustering and routing with the closest neighbor (ICR-CN) & GA
- Independent clustering and routing with the time window (ICR-TW) & GA

Phase 2 of the methodology is an improvement phase in which the GA is utilized. Considering the pool size, which is the number of initial feasible solutions in the pool, the algorithm tries to obtain the feasible solutions required in the specified number of replications. In our computational study, we first show the improvement gained by using the proposed GA in the second phase over the initial feasible solutions.

The algorithms in the solution methodology are implemented in C# language using Microsoft Visual Studio.12.0 on a personal computer with an Intel Core i7 CPU and 12 GB RAM.

### 5.3.1. Improvement by the Genetic Algorithm in Phase 2

In this part of the computational study, for testing the validity of the solution approaches, we solve 12 test problems totally with the number of customers (nodes), *n*, as 30, 50, 75, and 100 (Appendix B). For each *n*, we generate three problem instances through varying distances and times on the network, time window range, pickup and delivery demands, and other parameters. Each of the 12 test problems is solved without the improvement phase, that is, by only using the three clustering and routing heuristics. Thus, the solutions obtained are called the initial feasible solutions, which are then improved by GA in Phase 2.

Considering 20 initial feasible solutions in the pool (pool size), to generate these 20 initial feasible solutions for each problem instance, we generate 100 giant tours from which we intend to obtain 20 feasible solutions for the VRPSPD-STW-P. If we cannot obtain 20 initial feasible solutions among the 100 giant tours, we generate another 100 giant tours to obtain 20 initial feasible solutions. So we carry on this process for as many as 100 replications.

For each of the three clustering and routing heuristics (RCR, ICR-CN, ICR-TW), we record the minimum objective function value together with *m* (number of vehicles used) in these tables among the 20 initial feasible solutions as the best initial solution (Tables 5.5, 5.6, and 5.7). The initial feasible solutions are then improved by using the GA in Phase 2. For improving the 20 initial feasible solutions, GA runs for 100 iterations in each of which 100 new populations are generated, and the best result is reported. The best solution thus obtained is recorded in Tables 5.5, 5.6, and 5.7, together with the improvements (%) obtained by using GA in Phase 2. The improvement (%) is defined as:

[(best initial solution – best solution-GA) / best initial solution] *100%

We code each test problem as VTP*n*-*i* (*n* being the number of customer nodes on the network and *i* being the problem number), e.g., VTP30-1 indicates problem instance 1 of the three instances of the problem with 30 nodes. These problems are generated and tested using C# language of Microsoft Visual Studio.12.0.

In Tables 5.5, 5.6, and 5.7, the number of vehicles, *m*, used in the GA improved solution is also recorded for each problem instance.

The computational experiments indicate that the initial feasible solutions are improved by employing the proposed GA in Phase 2 of the methodology. The average percent improvements (API) in the best initial solutions are observed to be 8.9%, 8.2%, and 9.1%, respectively, for RCR, ICR-CN, ICR-TW heuristics.

Table 5.5. Improvements by the RCR&GA approach over the best initial solution by the RCR

| problem | No. of nodes | RCR | RCR&GA | *m* (vehicles used) | % Improvement with RCR&GA |
|---|---|---|---|---|---|
| | | Best initial solution | Best solution | | |
| VTP30-1 | 30 | 767,855 | 656,579 | 12 | 14.4 |
| VTP30-2 | 30 | 411,195 | 398,837 | 10 | 3.0 |
| VTP30-3 | 30 | 42,914 | 41,147 | 7 | 4.1 |
| VTP50-4 | 50 | 1,689,210 | 1,380,349 | 19 | 18.2 |
| VTP50-5 | 50 | 616,889 | 565,534 | 14 | 8.3 |
| VTP50-6 | 50 | 82,303 | 69,807 | 13 | 15.1 |
| VTP75-7 | 75 | 149,419 | 129,428 | 25 | 13.3 |
| VTP75-8 | 75 | 1,337,294 | 1,208,681 | 38 | 9.6 |
| VTP75-9 | 75 | 140,380 | 126,233 | 31 | 10.0 |
| VTP100-10 | 100 | 4,380,079 | 4,277,627 | 68 | 2.3 |
| VTP100-11 | 100 | 2,181,923 | 2,003,339 | 55 | 8.1 |
| VTP100-12 | 100 | 187,966 | 187,039 | 57 | 0.4 |
| API | | | | | 8.9 |

Table 5.6. Improvements by the ICR-CN&GA approach over the best initial solution by the ICR-CN

| problem | No. of nodes | ICR-CN | ICR-CN&GA | m (vehicles used) | % Improvement with ICR-CN&GA |
|---|---|---|---|---|---|
| | | Best initial solution | Best solution | | |
| VTP30-1 | 30 | 757,653 | 631,066 | 12 | 16.7 |
| VTP30-2 | 30 | 408,615 | 391,474 | 10 | 4.1 |
| VTP30-3 | 30 | 42,571 | 39,965 | 7 | 6.1 |
| VTP50-4 | 50 | 1,590,709 | 1,348,985 | 19 | 15.1 |
| VTP50-5 | 50 | 609,081 | 545,677 | 14 | 10.4 |
| VTP50-6 | 50 | 78,930 | 68,229 | 12 | 13.5 |
| VTP75-7 | 75 | 132,552 | 127,448 | 25 | 3.8 |
| VTP75-8 | 75 | 1,299,299 | 1,192,163 | 38 | 8.2 |
| VTP75-9 | 75 | 132,364 | 122,851 | 30 | 7.1 |
| VTP100-10 | 100 | 4,239,618 | 4,201,081 | 62 | 0.9 |
| VTP100-11 | 100 | 2,146,395 | 1,986,449 | 55 | 7.4 |
| VTP100-12 | 100 | 191,959 | 181,791 | 56 | 5.2 |
| API | | | | | 8.2 |

Table 5.7. Improvements by the ICR-TW&GA approach over the best initial solution by the ICR-TW

| problem | No. of nodes | ICR-TW | ICR-TW&GA | m (vehicles used) | % Improvement with ICR-TW&GA |
|---|---|---|---|---|---|
| | | Best initial solution | Best solution | | |
| VTP30-1 | 30 | 840,256 | 634,686 | 12 | 24.4 |
| VTP30-2 | 30 | 412,533 | 378,596 | 9 | 8.2 |
| VTP30-3 | 30 | 42,494 | 40,366 | 7 | 5.0 |
| VTP50-4 | 50 | 1,626,310 | 1,366,388 | 20 | 15.9 |
| VTP50-5 | 50 | 607,780 | 557,870 | 14 | 8.2 |
| VTP50-6 | 50 | 80,982 | 69,726 | 13 | 13.8 |
| VTP75-7 | 75 | 137,053 | 129,369 | 26 | 5.6 |
| VTP75-8 | 75 | 1,226,691 | 1,082,348 | 30 | 11.7 |
| VTP75-9 | 75 | 132,364 | 123,878 | 30 | 6.4 |
| VTP100-10 | 100 | 4,203,918 | 4,203,918 | 62 | 0 |
| VTP100-11 | 100 | 1,991,728 | 1,947,152 | 57 | 2.2 |
| VTP100-12 | 100 | 202,725 | 185,019 | 57 | 8.7 |
| API | | | | | 9.1 |

### 5.3.2. Comparison of the solution approaches

The final appraisal of the solution approaches rests on how close the solutions obtained by the proposed approaches get to the optimal solutions for the test problems under consideration. It is not always possible to obtain optimal solutions to challenging combinatorial problems. For this reason, comparisons are often made with the corresponding best integer solutions for the problems obtained by the exact solution methods within a specified computing time.

In this part of the computational study, we generate test problems with nodes, *n*, between 5 and 75 (Appendix C). For each *n*, we generate 10 instances completely randomly with different parameters. First, we obtain the best integer solution for VRPSPD-STW-P by CPLEX. We adopt the strategy of using CPLEX to run each test problem for a specified computing time, depending on the problem size, *n*.

The average deviation in the objective value is found between the best solution obtained by the proposed solution approaches and the corresponding best integer solution found by CPLEX. For more confident and stronger comparisons for the larger size problems, we implement CPLEX Optimization Studio in a computer with Intel Zeon 1260 Quad-Core CPU, 2.9 GHz, and 128 GB RAM. We attempt to solve the test problems by CPLEX in the allowed time ranges of 5 or 10 hours.

The corresponding computational results are shown in Tables 5.8 to 5.15. Each table includes the solutions and comparisons for a single size, *n*, for the problem. Tables 5.8 to 5.15 demonstrate the best solutions obtained by the three solution approaches for each problem. Furthermore, we compare the best solutions obtained by the three solution approaches against the best or optimal CPLEX solution. When we do not get an optimal solution for a problem by CPLEX due to the computer's limited storage capacity and unreasonable computing time, we limit the computing time by a specified time and interrupt the run at the end of this specified computing time, and then report both the optimality gap and the best integer solution obtained so far as an upper bound for the optimal solution of the problem.

The percent deviation between the best solution obtained by the solution approaches and the best or optimal solution obtained by CPLEX, is recorded as "%dev" in the tables, which is defined as:

$$\%\ \text{deviation} = [(\text{Solution approach's solution} - \text{CPLEX solution})/\text{CPLEX solution}] * 100$$

The average percent gap (APG), the average percent deviation (APD) and the number of vehicles used, *m*, are also shown in Tables 5.8 to 5.15. It is observed in these tables that APD and the %dev over all test problems except 50- and 75-node problems are less than 5.5 %.

Table 5.8. Comparison of the three solution approaches and the exact method:**5-node problems**

| Problem | RCR&GA | | ICR-CN&GA | | ICR-TW&GA | | Best solution of the three approaches | | | Solution by CPLEX | | | | %dev |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *m* | objective value | *m* | objective value | *m* | objective value | Best objective value | *m* | CPU time (min) | Optimal objective value | *m* | %Gap | CPU time (min) | |
| VTP5-1 | 1 | 6,427 | 2 | 8,016 | 2 | 7,884 | 6,427 | 1 | < 1 | 6,427 | 1 | 0 | 0.0041 | 0 |
| VTP5-2 | 1 | 44,805 | 1 | 45,825 | 1 | 48,206 | 44,805 | 1 | < 1 | 44,805 | 1 | 0 | 0.0028 | 0 |
| VTP5-3 | 1 | 55,698 | 2 | 62,759 | 2 | 66,274 | 55,698 | 1 | < 1 | 55,698 | 1 | 0 | 0.0028 | 0 |
| VTP5-4 | 1 | 50,189 | 2 | 70,257 | 1 | 55,608 | 50,189 | 1 | < 1 | 50,189 | 1 | 0 | 0.0040 | 0 |
| VTP5-5 | 1 | 48,192 | 2 | 70,949 | 1 | 61,331 | 48,192 | 1 | < 1 | 48,192 | 1 | 0 | 0.0030 | 0 |
| VTP5-6 | 1 | 6,982 | 1 | 7,845 | 2 | 8,336 | 6,982 | 1 | < 1 | 6,982 | 1 | 0 | 0.0028 | 0 |
| VTP5-7 | 2 | 67,580 | 2 | 67,885 | 2 | 67,580 | 67,580 | 2 | < 1 | 67,580 | 2 | 0 | 0.0040 | 0 |
| VTP5-8 | 1 | 5,297 | 1 | 6,130 | 1 | 6,061 | 5,297 | 1 | < 1 | 5,297 | 1 | 0 | 0.0051 | 0 |
| VTP5-9 | 1 | 6,282 | 1 | 6,581 | 1 | 6,909 | 6,282 | 1 | < 1 | 6,282 | 1 | 0 | 0.0059 | 0 |
| VTP5-10 | 1 | 5,744 | 1 | 7,014 | 2 | 7,080 | 5,744 | 1 | < 1 | 5,744 | 1 | 0 | 0.0029 | 0 |
| APG&APD | | | | | | | | | | | | 0 | | 0 |

*m*: vehicles used

Table 5.9. Comparison of the three solution approaches and the exact method:8-node problems

| Problem | RCR&GA | | ICR-CN&GA | | ICR-TW&GA | | Best solution of the three approaches | | | Solution by CPLEX | | | | %dev |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *m* | objective value | *m* | objective value | *m* | objective value | Best objective value | *m* | CPU time (min) | Optimal objective value | *m* | %Gap | CPU time (min) | |
| VTP8-1 | 1 | 71,105 | 2 | 84,805 | 2 | 84,766 | 71,105 | 1 | < 1 | 71,105 | 1 | 0 | 0.047 | 0 |
| VTP8-2 | 2 | 149,829 | 2 | 153,073 | 2 | 178,617 | 149,829 | 2 | < 1 | 146,166 | 2 | 0 | 0.040 | 2.5 |
| VTP8-3 | 2 | 85,704 | 2 | 88,324 | 2 | 96,237 | 85,704 | 2 | < 1 | 82,953 | 2 | 0 | 0.192 | 3.3 |
| VTP8-4 | 2 | 116,557 | 2 | 117,019 | 2 | 158,198 | 116,557 | 2 | < 1 | 116,557 | 2 | 0 | 0.019 | 0 |
| VTP8-5 | 1 | 69,578 | 2 | 81,199 | 2 | 88,569 | 69,578 | 1 | < 1 | 69,578 | 1 | 0 | 0.776 | 0 |
| VTP8-6 | 1 | 138,860 | 2 | 145,468 | 2 | 176,828 | 138,860 | 1 | < 1 | 133,429 | 2 | 0 | 0.0155 | 4.0 |
| VTP8-7 | 1 | 250,792 | 1 | 284,951 | 2 | 331,433 | 250,792 | 1 | < 1 | 250,792 | 1 | 0 | 00.95 | 0 |
| VTP8-8 | 4 | 408,599 | 4 | 414,261 | 4 | 409,296 | 408,599 | 4 | < 1 | 408,599 | 4 | 0 | 0.043 | 0 |
| VTP8-9 | 3 | 221,937 | 3 | 222,119 | 3 | 228,600 | 221,937 | 3 | < 1 | 221,937 | 3 | 0 | 0.026 | 0 |
| VTP8-10 | 3 | 371,332 | 2 | 379,051 | 3 | 404,037 | 371,332 | 3 | < 1 | 371,332 | 3 | 0 | 0.020 | 0 |
| APG&APD | | | | | | | | | | | | 0 | | 0.98 |

*m*: vehicles used

Table 5.10. Comparison of the three solution approaches and the exact method:10-node problems

| Problem | RCR&GA | | ICR-CN&GA | | ICR-TW&GA | | Best solution of the three approaches | | | Solution by CPLEX | | | | %dev |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $m$ | objective value | $m$ | objective value | $m$ | objective value | Best objective value | $m$ | CPU time (min) | Optimal objective value | $m$ | %Gap | CPU time (min) | |
| VTP10-1 | 2 | 102,139 | 2 | 106,587 | 2 | 108,062 | 102,139 | 2 | < 1 | 102,139 | 2 | 0 | 1.91 | 0 |
| VTP10-2 | 2 | 78,635 | 2 | 80,407 | 2 | 102,624 | 78,635 | 2 | < 1 | 76,254 | 2 | 0 | 3.13 | 3.1 |
| VTP10-3 | 2 | 267,125 | 2 | 252,759 | 2 | 264,337 | 252,759 | 2 | < 1 | 243,615 | 2 | 0 | 0.14 | 3.7 |
| VTP10-4 | 2 | 94,460 | 2 | 91,916 | 2 | 108,285 | 91,916 | 2 | < 1 | 89,395 | 2 | 0 | 10.89 | 2.8 |
| VTP10-5 | 2 | 109,207 | 2 | 129,540 | 2 | 127,994 | 109,207 | 2 | < 1 | 105,653 | 2 | 0 | 10.35 | 3.3 |
| VTP10-6 | 2 | 87,129 | 2 | 96,120 | 3 | 105,924 | 87,129 | 2 | < 1 | 83,850 | 2 | 0 | 93.13 | 3.9 |
| VTP10-7 | 2 | 12,021 | 2 | 13,100 | 2 | 13,990 | 12,021 | 2 | < 1 | 12,021 | 2 | 0 | 13.65 | 0 |
| VTP10-8 | 2 | 236,259 | 2 | 219,732 | 2 | 349,783 | 219,732 | 2 | < 1 | 219,732 | 2 | 0 | 0.07 | 0 |
| VTP10-9 | 2 | 90,338 | 2 | 100,320 | 2 | 112,392 | 90,338 | 2 | < 1 | 88,553 | 2 | 0 | 61.96 | 2.0 |
| VTP10-10 | 2 | 262,708 | 2 | 251,283 | 2 | 314,965 | 251,283 | 2 | < 1 | 251,283 | 2 | 0 | 0.06 | 0 |
| APG&APD | | | | | | | | | | | | 0 | | 1.88 |

$m$: vehicles used

Table 5.11. Comparison of the three solution approaches and the exact method:12-node problems

| Problem | RCR&GA | | ICR-CN&GA | | ICR-TW&GA | | Best solution of the three approaches | | | Solution by CPLEX | | | | %dev |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $m$ | objective value | $m$ | objective value | $m$ | objective value | Best objective value | $m$ | CPU time (min) | Optimal objective value | $m$ | %Gap | CPU time (min) | |
| VTP12-1 | 3 | 206,008 | 3 | 197,630 | 3 | 229,786 | 197,630 | 3 | < 5 | 193,938 | 3 | 0 | 1.35 | 1.9 |
| VTP12-2 | 2 | 387,935 | 2 | 333,406 | 3 | 434,586 | 333,406 | 2 | < 5 | 318,218 | 2 | 0 | 2.14 | 4.7 |
| VTP12-3 | 2 | 259,271 | 3 | 233,782 | 2 | 306,707 | 233,782 | 3 | < 5 | 232,100 | 3 | 0 | 0.78 | 0.7 |
| VTP12-4 | 3 | 447,772 | 3 | 395,722 | 3 | 472,380 | 395,722 | 3 | < 5 | 387,248 | 3 | 0 | 0.82 | 2.1 |
| VTP12-5 | 4 | 551,012 | 4 | 538,435 | 4 | 570,243 | 538,435 | 4 | < 5 | 535,004 | 4 | 0 | 259.24 | 0.6 |
| VTP12-6 | 2 | 219,334 | 2 | 190,985 | 3 | 240,903 | 190,985 | 2 | < 5 | 186,267 | 2 | 0 | 2.00 | 2.5 |
| VTP12-7 | 3 | 454,234 | 3 | 399,664 | 3 | 470,242 | 399,664 | 3 | < 5 | 391,699 | 3 | 0 | 2.36 | 2.0 |
| VTP12-8 | 3 | 150,859 | 3 | 137,554 | 3 | 159,137 | 137,554 | 3 | < 5 | 132,883 | 3 | 0 | 9.58 | 3.5 |
| VTP12-9 | 5 | 67,545 | 5 | 60,455 | 5 | 70,895 | 60,455 | 5 | < 5 | 60,455 | 5 | 0 | 6.68 | 0 |
| VTP12-10 | 3 | 315,106 | 3 | 274,647 | 3 | 319,028 | 274,647 | 3 | < 5 | 266,342 | 3 | 0 | 2.95 | 3.1 |
| APG&APD | | | | | | | | | | | | 0 | | 2.11 |

$m$: vehicles used

Table 5.12. Comparison of the three solution approaches and the exact method:20-node problems

| Problem | RCR&GA | | ICR-CN&GA | | ICR-TW&GA | | Best solution of the three approaches | | | Solution by CPLEX | | | | %dev |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *m* | objective value | *m* | objective value | *m* | objective value | Best objective value | *m* | CPU time (min) | Best objective value | *m* | %Gap | CPU time (min) | |
| VTP20-1 | 3 | 170,390 | 3 | 160,179 | 3 | 185,053 | 160,179 | 3 | < 10 | 154,994 | 3 | 5.29 | 300 | 3.3 |
| VTP20-2 | 3 | 90,653 | 3 | 95,685 | 3 | 97,515 | 90,653 | 3 | < 10 | 86,737 | 3 | 4.87 | 300 | 4.5 |
| VTP20-3 | 3 | 110,331 | 3 | 111,121 | 3 | 109,177 | 109,177 | 3 | < 10 | 103,351 | 3 | 5.40 | 300 | 5.6 |
| VTP20-4 | 6 | 62,932 | 5 | 59,908 | 6 | 64,363 | 59,908 | 5 | < 10 | 58,305 | 5 | 6.25 | 300 | 2.7 |
| VTP20-5 | 8 | 96,945 | 8 | 95,420 | 8 | 96,949 | 95,420 | 8 | < 10 | 91,679 | 9 | 7.95 | 300 | 4.0 |
| VTP20-6 | 4 | 125,682 | 4 | 121,219 | 4 | 125,520 | 121,219 | 4 | < 10 | 116,255 | 4 | 6.62 | 300 | 4.2 |
| VTP20-7 | 4 | 140,741 | 4 | 143,850 | 3 | 132,608 | 132,608 | 3 | < 10 | 126,573 | 3 | 5.11 | 300 | 4.7 |
| VTP20-8 | 3 | 177,568 | 3 | 167,948 | 3 | 185,938 | 167,948 | 3 | < 10 | 159,950 | 3 | 5.93 | 300 | 5.0 |
| VTP20-9 | 4 | 37,803 | 4 | 35,402 | 4 | 36,775 | 35,402 | 4 | < 10 | 35,352 | 4 | 4.22 | 300 | 0.1 |
| VTP20-10 | 6 | 88,379 | 7 | 90,174 | 6 | 90,670 | 88,379 | 6 | < 10 | 84,975 | 6 | 7.86 | 300 | 4.0 |
| APG&APD | | | | | | | | | | | | 5.95 | | 3.81 |

*m*: vehicles used

Table 5.13. Comparison of the three solution approaches and the exact method:30-node problems

| Problem | RCR&GA | | ICR-CN&GA | | ICR-TW&GA | | Best solution of the three approaches | | | Solution by CPLEX | | | | %dev |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *m* | objective value | *m* | objective value | *m* | objective value | Best objective value | *m* | CPU time (min) | Best objective value | *m* | %Gap | CPU time (min) | |
| VTP30-1 | 6 | 295,720 | 6 | 293,731 | 6 | 294,298 | 293,731 | 6 | < 10 | 292,528 | 6 | 10.34 | 300 | 0.4 |
| VTP30-2 | 7 | 191,120 | 6 | 176,428 | 7 | 191,206 | 176,428 | 6 | < 10 | 168,269 | 6 | 9.24 | 300 | 4.8 |
| VTP30-3 | 5 | 196,833 | 5 | 194,583 | 5 | 201,585 | 194,583 | 5 | < 10 | 192,850 | 5 | 9.18 | 300 | 0.8 |
| VTP30-4 | 8 | 56,380 | 9 | 55,153 | 8 | 57,184 | 55,153 | 9 | < 10 | 52,084 | 8 | 12.77 | 300 | 5.8 |
| VTP30-5 | 6 | 226,177 | 6 | 229,887 | 6 | 236,208 | 226,177 | 6 | < 10 | 218,254 | 6 | 11.63 | 300 | 3.6 |
| VTP30-6 | 5 | 150,356 | 5 | 151,828 | 5 | 144,221 | 144,221 | 5 | < 10 | 138,823 | 5 | 8.06 | 300 | 3.8 |
| VTP30-7 | 6 | 95,104 | 6 | 91,272 | 6 | 95,165 | 91,272 | 6 | < 10 | 89,237 | 6 | 10.19 | 300 | 2.2 |
| VTP30-8 | 4 | 339,951 | 4 | 325,093 | 4 | 363,446 | 325,093 | 4 | < 10 | 309,998 | 3 | 9.80 | 300 | 4.8 |
| VTP30-9 | 5 | 62,276 | 5 | 59,233 | 5 | 61,297 | 59,233 | 5 | < 10 | 56,108 | 4 | 9.43 | 300 | 5.5 |
| VTP30-10 | 7 | 110,104 | 7 | 109,864 | 7 | 113,516 | 109,864 | 7 | < 10 | 109,308 | 7 | 8.32 | 300 | 0.5 |
| APG&APD | | | | | | | | | | | | 9.89 | | 3.22 |

*m*: vehicles used

Table 5.14. Comparison of the three solution approaches and the exact method:50-node problems

| Problem | RCR&GA | | ICR-CN&GA | | ICR-TW&GA | | Best solution of the three approaches | | | Solution by CPLEX | | | | %dev** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *m* | objective value | *m* | objective value | *m* | objective value | Best objective value | *m* | CPU time (min) | Best objective value | *m* | %Gap | CPU time (min) | |
| VTP50-1 | 13 | 164,226 | 13 | 156,505 | 12 | 164,436 | 156,505 | 13 | < 20 | 181,658 | 13 | 37.20 | 600 | -13.8 |
| VTP50-2 | 8 | 354,150 | 9 | 339,539 | 8 | 346,642 | 339,539 | 9 | < 20 | 397,859 | 8 | 31.38 | 600 | -14.6 |
| VTP50-3 | 9 | 439,337 | 8 | 402,571 | 8 | 424,562 | 402,571 | 8 | < 20 | 465,511 | 7 | 28.87 | 600 | -13.5 |
| VTP50-4 | 15 | 154,319 | 12 | 149,484 | 13 | 153,509 | 149,484 | 12 | < 20 | 172,643 | 14 | 25.51 | 600 | -13.4 |
| VTP50-5 | 18 | 239,925 | 16 | 236,459 | 18 | 250,434 | 236,459 | 16 | < 20 | 280,167 | 15 | 35.44 | 600 | -15.6 |
| VTP50-6 | 7 | 307,575 | 6 | 293,181 | 7 | 296,302 | 293,181 | 6 | < 20 | 335,954 | 8 | 19.88 | 600 | -12.7 |
| VTP50-7 | 8 | 345,945 | 7 | 317,860 | 8 | 350,846 | 317,860 | 7 | < 20 | 362,112 | 8 | 26.15 | 600 | -12.2 |
| VTP50-8 | 9 | 51,321 | 10 | 47,678 | 11 | 50,070 | 47,678 | 10 | < 20 | 63,904 | 10 | 42.16 | 600 | -25.3 |
| VTP50-9 | 13 | 149,947 | 14 | 147,352 | 14 | 154,300 | 147,352 | 14 | < 20 | 173,051 | 15 | 36.35 | 600 | -14.8 |
| VTP50-10 | 8 | 209,294 | 8 | 200,775 | 7 | 206,962 | 200,775 | 8 | < 20 | 272,676 | 10 | 41.69 | 600 | -26.3 |
| APG&APD | | | | | | | | | | | | 32.46 | | -16.2 |

*m*: vehicles used

%dev**: Best solution is better than CPLEX solution.

Table 5.15. Comparison of the three solution approaches and the exact method:75-node problems

| Problem | RCR&GA | | ICR-CN&GA | | ICR-TW&GA | | Best solution of the three approaches | | | Solution by CPLEX | | | | %dev** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *m* | objective value | *m* | objective value | *m* | objective value | Best objective value | *m* | CPU time (min) | Best objective value | *m* | %Gap | CPU time (min) | |
| VTP75-1 | 20 | 258,994 | 18 | 239,950 | 21 | 254,233 | 239,950 | 18 | < 30 | 340,813 | 22 | 56.25 | 600 | -29.5 |
| VTP75-2 | 13 | 486,061 | 12 | 484,458 | 13 | 485,989 | 484,458 | 12 | < 30 | 565,905 | 13 | 38.23 | 600 | -14.3 |
| VTP75-3 | 13 | 642,509 | 12 | 605,678 | 13 | 623,079 | 605,678 | 12 | < 30 | 721,793 | 14 | 40.87 | 600 | -16.0 |
| VTP75-4 | 23 | 266,939 | 22 | 254,599 | 23 | 262,226 | 254,599 | 22 | < 30 | 300,909 | 19 | 42.35 | 600 | -15.3 |
| VTP75-5 | 10 | 249,066 | 10 | 226,060 | 9 | 255,285 | 226,060 | 10 | < 30 | 290,695 | 9 | 50.98 | 600 | -22.2 |
| VTP75-6 | 11 | 441,824 | 11 | 415,718 | 10 | 434,844 | 415,718 | 11 | < 30 | 471,542 | 10 | 37.59 | 600 | -11.8 |
| VTP75-7 | 13 | 87,790 | 12 | 82,223 | 13 | 89,128 | 82,223 | 12 | < 30 | 105,052 | 8 | 44.63 | 600 | -21.7 |
| VTP75-8 | 14 | 76,334 | 15 | 70,911 | 16 | 76,710 | 70,911 | 15 | < 30 | 84,786 | 12 | 37.42 | 600 | -16.3 |
| VTP75-9 | 16 | 106,723 | 16 | 98,144 | 18 | 105,404 | 98,144 | 16 | < 30 | 123,157 | 14 | 48.19 | 600 | -20.3 |
| VTP75-10 | 17 | 175,340 | 17 | 162,691 | 16 | 170,842 | 162,691 | 17 | < 30 | 213,344 | 14 | 38.08 | 600 | -23.7 |
| APG&APD | | | | | | | | | | | | 43.45 | | -19.1 |

*m*: vehicles used

%dev**: Best solution is better than CPLEX solution.

For all problems of $n$=5 nodes and most of the problems of $n$=8 nodes, the solution approach, RCR&GA, provides optimal solutions. These results can be a good test for the reliability of our solution methodology, and the solution times are quite comparable with the solution times required for the optimal solutions by CPLEX.

For the problems of $n$=10 nodes, the % deviation is in the range of 0% to 3.7%, 1.81% being the APD over 10 instances. The CPU time of the proposed solution approach providing the best solution is observed to be less than 1 minute for all problem instances, while it even rises to more than 1 hour for some problem instances by the CPLEX solver.

For the problems of $n$=12 nodes, the % deviation is in the range of 0% to 4.5%, 2.03% being the APD over 10 instances. The CPU time of the proposed solution approach (ICR-CN&GA) that provided the best solution for all instances with $n$=12 nodes is less than 5 minutes, while it changes in the range of 47 seconds to more than 4 hours with the CPLEX solver.

For the problems of $n$=20, the % deviation is in the range of 0.1% to 4.7%, 3.64% being the APD over 10 instances. The CPU time of the proposed solution approaches that provided the best solution is less than 10 minutes. The solutions by the CPLEX solver are the best feasible solutions obtained in the allowed 5 hours of computing time.

For the problems of $n$=30, the % deviation is in the range of 0.4% to 5.5%, 3.10% being the APD over 10 instances. As in the problems of $n$=20, again the solution approach (ICR-CN&GA) provided the best solution for the majority of the instances. The CPU time of the proposed solution approaches that provided the best solution is less than 10 minutes. The CPLEX solutions are the best feasible solutions obtained in the allowed 5 hours of computing time.

In problems of $n$=20 and $n$=30 nodes, the CPLEX solver cannot provide the optimal solution. The %gap between the best integer solution and best bound (relaxation problem solution) found by the CPLEX solver within 5 hours of computing time is less than 5%. This amount of gap indicates that the CPLEX best solution is close to the optimal solution.

For problem sets with *n*=50 and *n*=75 nodes, the results show that the solutions of the proposed (ICR-CN&GA) approach are better than the solutions found by the CPLEX solver. CPLEX returns inferior solutions for all problem instances in the allowed computing time of 10 hours. The proposed solution approach, ICR-CN&GA, gives better solutions with APDs of 19.5% and 23.9%, respectively, for *n*=50 and *n*=75.

The CPLEX solver takes hours to find the given feasible solutions in most of the cases. From the above test results, we find that the solution approaches in our proposed methodology can solve VRPSPD-STW-P efficiently and returns a satisfactory solution. The efficiency of our methodology makes it suitable for solving a real case, which is usually large in terms of the number of nodes.

Finally, we compare the three solution approaches among themselves. We use the same test problems in Section 5.2.2 with the number of nodes *n*=20, 30, 50, and 75 (Appendix D). In Tables 5.16, 5.17, 5.18, and 5.19, we show the best solution and the percent deviations among the solutions of the three proposed approaches. The percent deviation of a solution with a solution approach from the best solution is given by:

$$\% \text{ deviation} = [(\text{Solution} - \text{Best solution}) / \text{Best solution}] * 100$$

The average percent deviations (APD) is presented at the bottom of the tables. Table 5.20, as a summary table, presents the APD values of the three solution approaches for the problem sets.

Table 5.16. Best solution and % deviation of the solution approaches from the best - $n$=20

| Problem | Best solution | % deviation from the best solution | | |
|---|---|---|---|---|
| | | RCR&GA | ICR-CN&GA | ICR-TW&GA |
| VTP20-1 | 160,179 | 5.9 | 0 | 13.4 |
| VTP20-2 | 90,653 | 0 | 5.2 | 7.0 |
| VTP20-3 | 109,177 | 1.0 | 1.7 | 0 |
| VTP20-4 | 59,908 | 4.8 | 0 | 6.9 |
| VTP20-5 | 95,420 | 1.5 | 0 | 1.5 |
| VTP20-6 | 121,219 | 3.5 | 0 | 3.4 |
| VTP20-7 | 132,608 | 5.7 | 7.8 | 0 |
| VTP20-8 | 167,948 | 5.4 | 0 | 9.6 |
| VTP20-9 | 35,402 | 6.3 | 0 | 3.7 |
| VTP20-10 | 88,379 | 0 | 1.9 | 2.5 |
| Number of best solutions found | | 2 | 6 | 2 |
| APD | | 3.41 | 1.66 | 4.80 |
| Min APD | ICR-CN&GA with 1.66% | | | |
| Max APD | ICR-TW&GA with 4.80% | | | |

Table 5.17. Best solution and % deviation of the solution approaches from the best - $n$=30

| Problem | Best solution | % deviation from the best solution | | |
|---|---|---|---|---|
| | | RCR&GA | ICR-CN&GA | ICR-TW&GA |
| VTP30-1 | 293,731 | 0.67 | 0 | 0.19 |
| VTP30-2 | 176,428 | 7.6 | 0 | 7.7 |
| VTP30-3 | 194,583 | 1.1 | 0 | 3.4 |
| VTP30-4 | 55,153 | 2.1 | 0 | 3.5 |
| VTP30-5 | 226,177 | 0 | 1.6 | 4.2 |
| VTP30-6 | 144,221 | 4.0 | 5.0 | 0 |
| VTP30-7 | 91,272 | 4.0 | 0 | 4.0 |
| VTP30-8 | 325,093 | 4.3 | 0 | 10.5 |
| VTP30-9 | 59,233 | 4.8 | 0 | 3.3 |
| VTP30-10 | 109,864 | 0.21 | 0 | 3.2 |
| Number of best solutions found | | 1 | 8 | 1 |
| APD | | 2.87 | 0.66 | 3.99 |
| Min APD | ICR-CN&GA with 0.66% | | | |
| Max APD | ICR-TW&GA with 3.99% | | | |

Table 5.18. Best solution and % deviation of the solution approaches from the best - $n$=50

| Problem | Best solution | % deviation from the best solution | | |
|---|---|---|---|---|
| | | RCR&GA | ICR-CN&GA | ICR-TW&GA |
| VTP50-1 | 156,505 | 4.7 | 0 | 4.8 |
| VTP50-2 | 339,539 | 4.1 | 0 | 2.0 |
| VTP50-3 | 402,571 | 8.3 | 0 | 5.1 |
| VTP50-4 | 149,484 | 3.1 | 0 | 2.6 |
| VTP50-5 | 236,459 | 1.4 | 0 | 5.5 |
| VTP50-6 | 293,181 | 4.6 | 0 | 1.0 |
| VTP50-7 | 317,860 | 8.1 | 0 | 9.4 |
| VTP50-8 | 47,678 | 7.0 | 0 | 4.7 |
| VTP50-9 | 147,352 | 1.7 | 0 | 4.5 |
| VTP50-10 | 200,775 | 4.0 | 0 | 2.9 |
| Number of best solutions found | | 0 | 10 | 0 |
| APD | | 4.70 | 0 | 4.25 |
| Min APD | ICR-CN&GA with 0% | | | |
| Max APD | RCR&GA with 4.70% | | | |

Table 5.19. Best solution and % deviation of the solution approaches from the best - $n$=75

| Problem | Best solution | % deviation from the best solution | | |
|---|---|---|---|---|
| | | RCR&GA | ICR-CN&GA | ICR-TW&GA |
| VTP75-1 | 239,950 | 7.3 | 0 | 5.6 |
| VTP75-2 | 484,458 | 0.32 | 0 | 0.31 |
| VTP75-3 | 605,678 | 5.7 | 0 | 2.7 |
| VTP75-4 | 254,599 | 4.6 | 0 | 2.9 |
| VTP75-5 | 226,060 | 9.2 | 0 | 11.4 |
| VTP75-6 | 415,718 | 5.9 | 0 | 4.3 |
| VTP75-7 | 82,223 | 6.3 | 0 | 7.7 |
| VTP75-8 | 70,911 | 7.1 | 0 | 7.5 |
| VTP75-9 | 98,144 | 8.0 | 0 | 6.8 |
| VTP75-10 | 162,691 | 7.2 | 0 | 4.7 |
| Number of best solutions found | | 0 | 10 | 0 |
| APD | | 6.16 | 0 | 5.39 |
| Min APD | ICR-CN&GA with 0% | | | |
| Max APD | RCR&GA with 6.16% | | | |

Table 5.20. Summary of the APD values (%)

| Nodes, *n* | RCR&GA | ICR-CN&GA | ICR-TW&GA |
|---|---|---|---|
| 20 | 3.41 | 1.66 | 4.80 |
| 30 | 2.87 | 0.66 | 3.99 |
| 50 | 4.70 | 0 | 4.25 |
| 75 | 6.16 | 0 | 5.39 |
| Average of APD | 4.28 | 0.58 | 4.60 |

The computational results show that our proposed solution approach, ICR-CN&GA provides better solutions in most of the test problems compared to RCR&GA and ICR-TW&GA. In comparing the other two solution approaches with each other, it is seen that RCR&GA approach outperforms ICR-TW&GA for problems with $n$=20 and $n$=30 problems. However, the reverse is observed for problems of larger sizes with $n$=50 and $n$=75 nodes.

### 5.3.3. Sensitivity analysis

In the first analysis, we run a series of test problems with $n$=30, 50, and 75 nodes (Appendix E), in which the minimum preferred quality levels (freshness) are changed. Table 5.21 presents the results of the test problems by implementing the best approach ICR-CN&GA. The objective values and the number of vehicles, under the minimum preferred quality levels, are also given in Figures 5.7 and 5.8.

Table 5.21. Solutions under different quality levels

| Problem | $q$ | $m$ | Objective value | Problem | $m$ | Objective value | Problem | $m$ | Objective value |
|---|---|---|---|---|---|---|---|---|---|
| VTP30 | 0.1 | 4 | 209,962 | VTP50 | 8 | 327,934 | VTP75 | 11 | 467,930 |
| VTP30 | 0.3 | 4 | 210,968 | VTP50 | 8 | 332,516 | VTP75 | 11 | 469,917 |
| VTP30 | 0.5 | 4 | 214,884 | VTP50 | 10 | 349,155 | VTP75 | 14 | 502,508 |
| VTP30 | 0.7 | 4 | 217,132 | VTP50 | 27 | 640,208 | VTP75 | 37 | 875,665 |
| VTP30 | 0.9 | - | - | VTP50 | - | - | VTP75 | - | - |

$m$: (number of vehicles used)

The results show that the objective value and the total number of vehicles needed for finding a feasible solution increase when the minimum preferred quality level increases. Also, for low-quality levels, the number of vehicles does not change, but with the increase in the quality levels, $q \geq 0.50$, the number of vehicles increases sharply. In Table 5.21 on the last row with $q$=0.9, it should also be noted that considering travel times and also start of time window at customers, delivering products to customers with the preferred quality level of $q$=0.9 turns out to be impossible and hence problems are infeasible for this level of preferred quality.

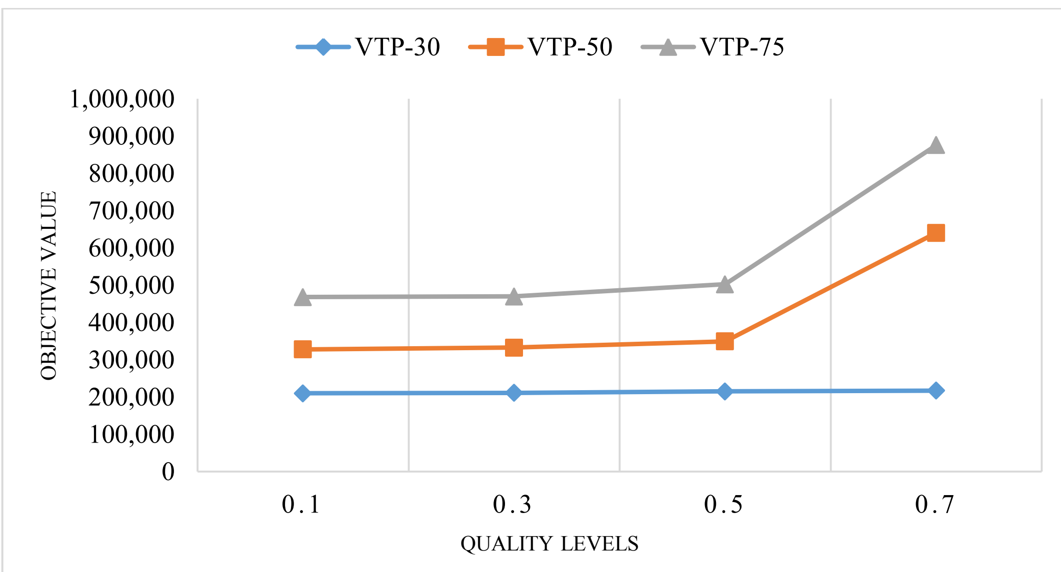



Figure 5.7. Objective values under different quality levels

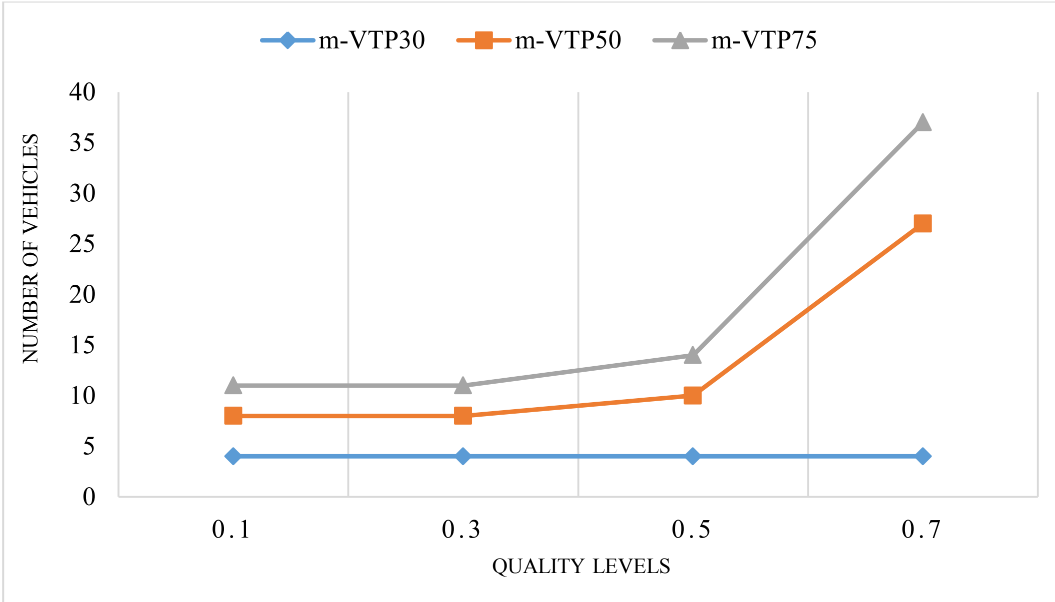



Figure 5.8. Number of vehicles under different quality levels

In the second analysis, we consider problems with $n$=30 and 50 nodes (Appendix F) to observe the effect of the percentage of customers with time window requirements, including TW(0%), TW(20%), TW(40%), TW(60%), TW(80%), and TW(100%). For example, TW(0%) refers to no customers with time window requirement, while TW(40%) refers to 40% of customers having time window requirements. The effect of these changes on the objective values is summarized in Table 5.22. Figures 5.9 and 5.10 show that the objective value increases when the percentage of customers with time window requirements increases, so does the number of vehicles after a threshold %value.

Table 5.22. Objective values under different % of customers with time windows

| Problem | *TW* (%) | *m* | Objective value | Problem | *TW* (%) | *m* | Objective value |
|---|---|---|---|---|---|---|---|
| VTP30 | 0 | 2 | 123,518 | VTP50 | 0 | 5 | 50,785 |
| VTP30 | 20 | 2 | 124,852 | VTP50 | 20 | 5 | 52,838 |
| VTP30 | 40 | 2 | 147,439 | VTP50 | 40 | 5 | 56,495 |
| VTP30 | 60 | 2 | 166,863 | VTP50 | 60 | 6 | 65,853 |
| VTP30 | 80 | 3 | 202,639 | VTP50 | 80 | 6 | 69,217 |
| VTP30 | 100 | 3 | 216,255 | VTP50 | 100 | 6 | 71,433 |

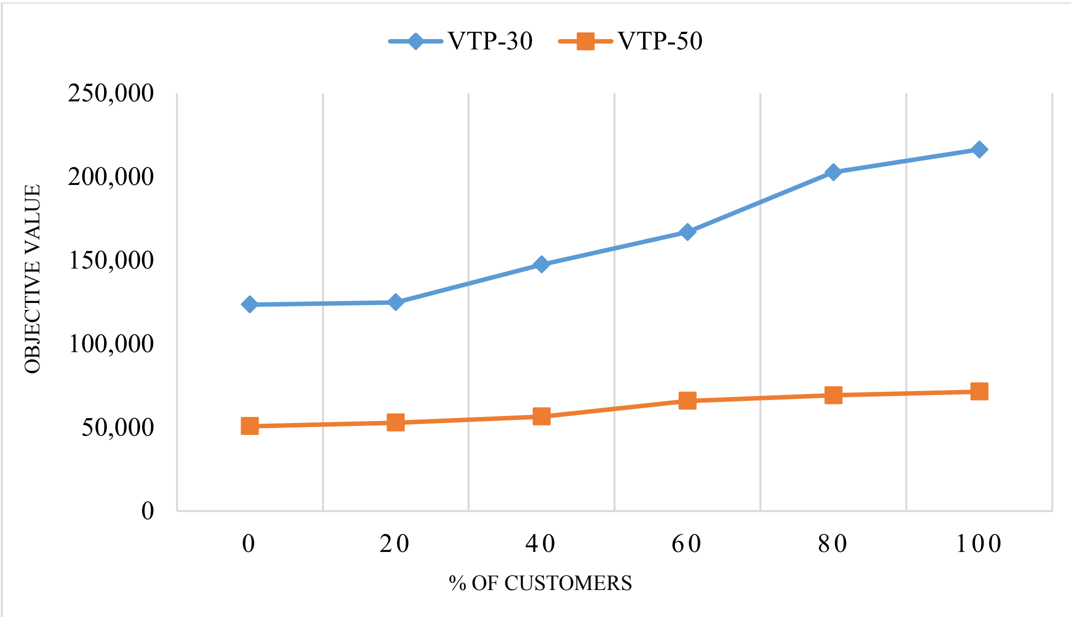



Figure 5.9. Objective values vs % of customers with time windows

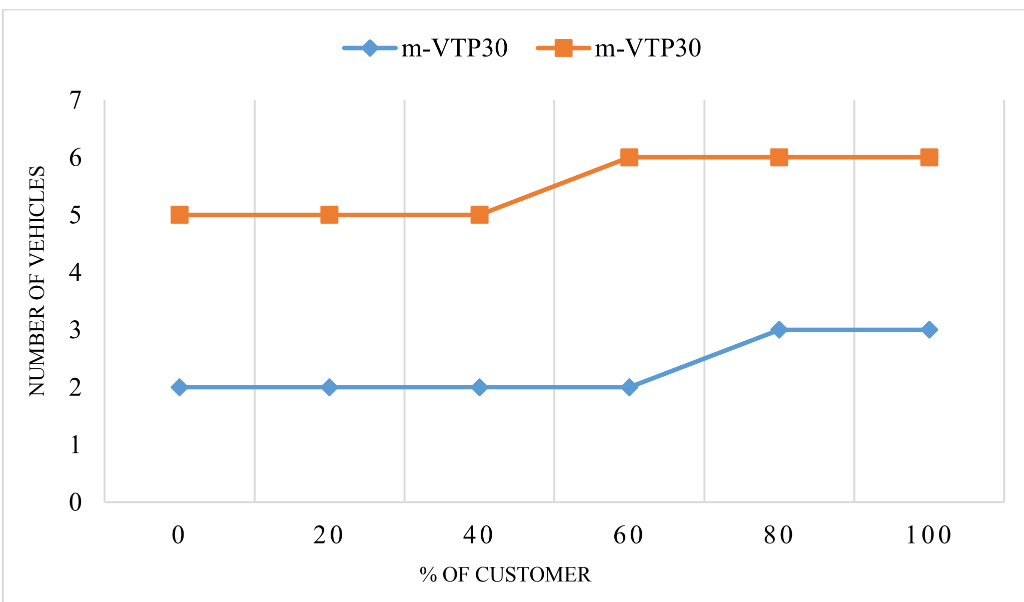



Figure 5.10. Number of vehicles vs % of customers with time windows

If the time windows are too large, then nothing changes even with all customers having time windows. What is more significant is how tight (narrow) the time windows are.

In the third analysis, problems with $n$=30 and 50 nodes (Appendix G) are considered for conducting sensitivity analysis in terms of the tightness of time window ranges. Assume different %'s between lower and upper bounds of time window ranges, including $ab$(20%), $ab$(50%), $ab$(100%), $ab$(150%), and $ab$(200%); e.g., $ab$(20%) refers to $b_i = a_i + 0.2a_i$ for all nodes, then the range is said to be narrow (tight); and $ab$(150%) refers to $b_i = a_i + 1.5a_i$, then the range is said to be wide (loose). The impact of these changes on the objective value is displayed in Figures 5.11 and 5.12, and summarized in Table 5.23. Figures 5.11 and 5.12 show that the objective value decreases when the range of time windows increases, that is, the range of time window is loose.

Table 5.23. Objective values under time window ranges

| Problem | $ab$% | $m$ | Objective value | Problem | $ab$% | $m$ | Objective value |
|---|---|---|---|---|---|---|---|
| VTP30 | 20 | 7 | 199,412 | VTP50 | 20 | 16 | 252,594 |
| VTP30 | 50 | 7 | 192,224 | VTP50 | 50 | 15 | 234,410 |
| VTP30 | 100 | 7 | 186,642 | VTP50 | 100 | 17 | 229,218 |
| VTP30 | 150 | 7 | 185,353 | VTP50 | 150 | 15 | 228,905 |
| VTP30 | 200 | 7 | 183,220 | VTP50 | 200 | 15 | 226,355 |

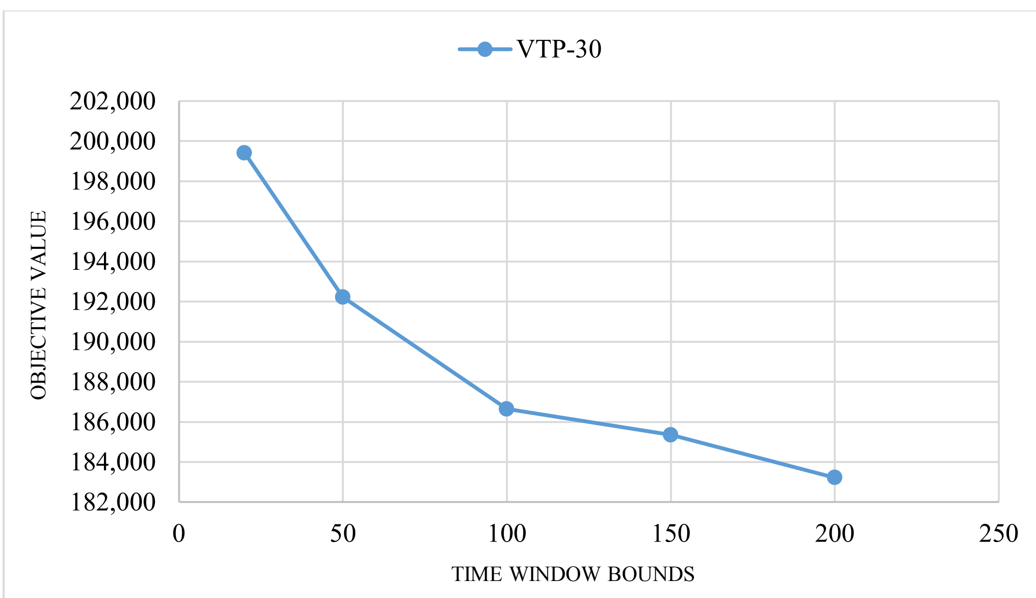



Figure 5.11. Objective values vs time window ranges- 30 nodes

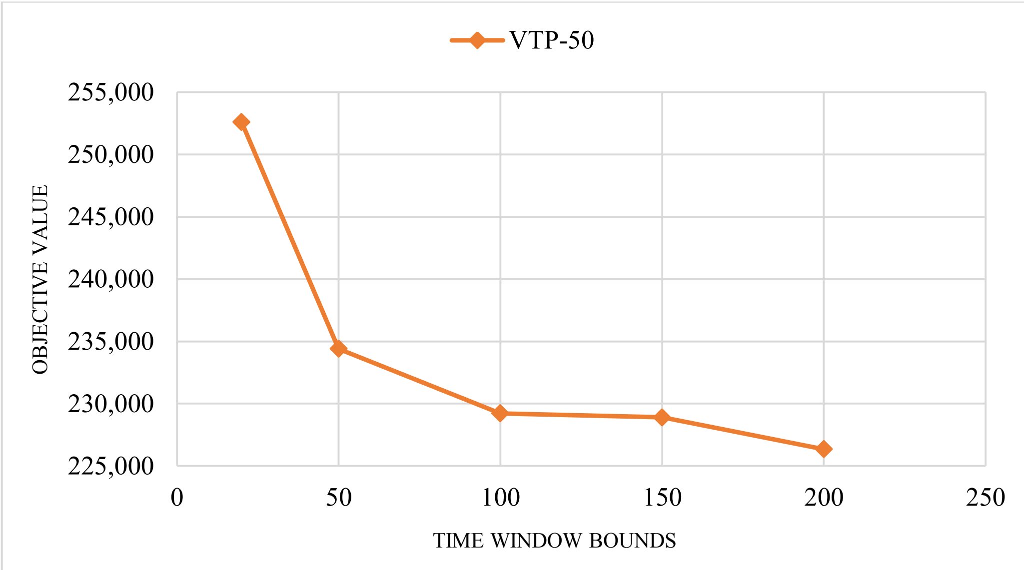



Figure 5.12. Objective values vs time window ranges- 50 nodes

Having compared the efficiency of our approach on different problem sizes, we observe that, in particular, for a large-size problem instance, i.e., 50 nodes, the computational time for obtaining a satisfactory solution with our approach is within 30 minutes, and that the solutions are much better than CPLEX solutions obtained within 10 hours running time.

One of our solution approaches, ICR-CN&GA, outperforms the other two solution approaches, RCR&GA and ICR-TW&GA, for most of the generated problem instances. Observing the details of the solutions which are the best solutions among the solutions provided by the three solution approaches proposed, it is seen that the best solutions end up with a less number of vehicles in the fleet and a larger number of customers served by each vehicle.

All in all, the results show that the methods for generating initial solutions in Phase 1 and the GA approach for solution improvement in Phase 2 are effective, and provide optimal or near-optimal solutions for the problems tested in a reasonable amount of time.

# CHAPTER 6

## CONCLUSIONS AND FURTHER RESEARCH ISSUES

In this study, we consider a sustainable supply chain for food products and attempt to minimize the distribution costs and impacts of supply chain operations on the environment and society. Sustainability is an essential issue in the perishable food supply chain. We take the sustainability of the transportation system into account through minimizing energy consumption in distribution and waste management as well, including collecting end-of-life (EOL) products, waste materials from customers by reverse flow for remanufacturing, reusing, and recycling.

Reverse product flow is a relevant contribution to the sustainability of distribution systems. One of the approaches towards sustainability and green manufacturing and logistics is to collect end-of-life products from customers for either reuse or proper disposal. We study the process of collecting end-of-life products as reverse logistics networks and satisfying the pickup demands of customers. The reduction of the amount of waste produced and energy consumed in distribution serves the economic and environmental aspects of sustainability.

Our study is concerned with the short-term planning of distribution and mostly dealing with deliveries/pickups to/from several customers. Considering sustainability in the food supply chains, therefore, we focus on the betterment of both economic and environmental aspects in a perishable food supply chain.

Vehicle routing and scheduling for perishable goods is integrated into a unified framework applicable to fields like food, vegetables, flowers, and even livestock. The problem aims to deliver food products with high quality as much as possible, at least as required by customers, and collect end-of-life products from customers with both delivery and pickup demand.

In our approach to this problem, we investigate an extension of the well-known vehicle routing problem (VRP) with the food-specific additional characteristics that include scheduling and routing of vehicles for perishable food products. We present a mixed-integer linear programming model for this vehicle routing problem with simultaneous pickup and delivery and soft time windows for perishable food supply chains (VRPSPD-STW-P).

The VRPSPD-STW-P is a combinatorial optimization problem, and it is NP-hard. Therefore, exact approaches to solve this problem become inefficient in general; and can only solve relatively small-size problems. Even for the small size problems with only a few nodes, solving the VRPSPD-STW-P using optimization software CPLEX takes a very long time and remains to be challenging, and as for the larger size real-life problems, the exact methods employed by CPLEX cannot provide even a feasible solution in a reasonable time.

We obtain CPLEX solutions for comparison purposes and end up with the best integer solution (best bound) within the pre-defined computational time. Several initial runs validate our MILP model, and the computational results indicate that our model is effective and efficient.

In our methodology we propose some solution approaches that are based on heuristic algorithms to solve our problem that we call VRPSPD-STW-P. Our proposed methodology mainly includes clustering of nodes, routing, genetic algorithm, and feasibility checking techniques to obtain near-optimal or optimal solutions for the VRPSPD-STW-P in two phases, obtaining initial feasible solutions in the first phase and improving them in the second phase. In the second phase, we adopt a genetic algorithm (GA) to improve the initial feasible solutions and determine the best solution that has the minimum cost at the end of the GA.

The basic idea in the methodology is decomposing the original problem with a great number of nodes in its network into a required number of smaller networks, which corresponds to generating a giant tour of all nodes and then obtaining $m$ smaller routes as a result of clustering. After obtaining $m$ many clusters (sub-networks), the methodology goes on with routing heuristics by relaxing all constraints of the problem to obtain the routes within these smaller networks (sub-networks). We develop three procedures for the routing of each cluster. Three different solution approaches only differentiated by the type of clustering and routing heuristic are as follows:

- Random clustering and routing (RCR)
- Independent clustering and routing with the closest neighbor (ICR-CN)
- Independent clustering and routing with the time window (ICR-TW)

At this point of the methodology, the solution obtained is just a solution for the $m$TSP problem. Then each route of $m$TSP is checked individually for feasibility incorporating the distance traveled by the vehicle in its tour and vehicle capacity limitation. If the solution is found to be feasible, then the solution is a feasible solution for TSPSPD, which is then passed to the feasibility check for the soft time windows and also quality level restrictions of customers of the route. If the solution is found to be feasible, then it turns out to be a feasible solution for the TSPSPD-STW-P. If, at the end of these feasibility checks, all $m$ routes are found to be feasible, then generated the $m$ routes are solutions for mTSPSPD-STW-P which is a feasible solution for the original problem VRPSPD-STW-P. Otherwise, the solution the $m$TSP solution is infeasible for the original problem VRPSPD-STW-P.

Therefore, the initial feasible solution for the VRPSPD-STW-P is obtained by combining the $m$ feasible routes of $m$TSPSPD-STW-P. Thus, the feasible solutions are improved by generating improved routes through the GA in the second phase of the methodology. Using three different vehicle routing heuristics in the clusters of nodes, we end up with three solution approaches in our methodology as follows:

- RCR&GA
- ICR-CN&GA
- ICR-TW&GA

We conduct several computational experiments to evaluate the performance of the proposed approaches with several problem instances generated for VRPSPD-STW-P. The results of the three proposed approaches are compared with each other and against the result provided by CPLEX. Our solution methodology yields very promising solutions in a much less computational time than optimal or feasible solutions generated by the exact solution procedure. Computational results show that our proposed heuristic methods for generating the initial solutions and the GA for solution improvement are effective and can obtain near-optimal or optimal solutions for the problems tested in polynomial computational times.

In summary, this study shows how the crucial characteristics as perishability in the safe delivery of food can appropriately be considered in formulating vehicle routing problems with pickup and delivery, and time-window constraints. The proposed methodology provides useful tools that may enable operators to make effective delivery and pickup decisions under the specified quality levels and late arrival penalties by assessing the impact of delivery times, food spoilage, and time windows on vehicle routing and the resultant operating costs.

If the quality of the products delivered is required to be higher, consequently higher distribution costs are incurred. So the number of vehicles turns out to be a critical factor in VRPSPD-STW-P in meeting the product’s required quality levels when delivered to the customers. In addition to using the owned vehicles, the planners may use more vehicles by renting additional vehicles to transport the perishable product.

When we compare our study against the similar studies in the literature, in terms of modeling characteristics, that is, the variant of VRP addressed, the proposed solution approaches including the problem instances tested, our work seems to be the first study that considers perishable food products distribution in a VRPSPD framework with soft time windows.

Further research could focus on applying the proposed approaches to real-world instances in a wide range of food industries. Next to practical outcomes in terms of decision support, this application may allow us further to study the performance of the model in different settings.

In the future study, different kinds of deterioration and more advanced quality decay models, such as nonlinear or exponential nature, may be considered in the VRPSPD-STW-P.

A further improvement in the study can also include the consideration of factors like time-dependent travel-times, and time-varying temperatures during the day. The temperature changes can affect the energy consumed by vehicles (for keeping the internal temperature of vehicles cool), so as an extension of the study, we can add cost of energy consumed by vehicles for time-varying temperatures.

There can be exploratory research into the freshness objective function that evaluates other indicators. In order to understand the "total cost-freshness" trade-off, a multi-objective model can be proposed to minimize the total distribution cost and maximize the freshness.

In this research, we consider a single product or a single family of products with a single depot in the distribution network. Future work could also include multiple products and multiple depots. This, however, has to consider different quality degradation models for the products. Furthermore, interactions among the products might have to be taken into account, as it is, for instance, in fruits and vegetables.
Aggregation of heuristics and simplification techniques in our solution methods may be developed. Thus the next step of research can include further development of our

solution methodology to incorporate other metaheuristics. Different metaheuristics such as tabu search and simulated annealing can be used for generating initial solutions and solution improvements.

**REFERENCES**


Ageron, B., Gunasekaran, A., & Spalanzani, A. (2012). Sustainable supply management: An empirical study. *International Journal of Production Economics*, *140*(1), 168-182.

Ahuja, R. K., Huang, W., Romeijn, H. E., & Morales, D. R. (2007). A heuristic approach to the multi-period single-sourcing problem with production and inventory capacities and perishability constraints. *INFORMS Journal on Computing*, *19*(1), 14-26.

Ahumada, O., & Villalobos, J. R. (2009). Application of planning models in the agri-food supply chain: A review. *European Journal of Operational Research*, *196*(1), 1-20.

Ahumada, O., & Villalobos, J. R. (2011). A tactical model for planning the production and distribution of fresh produce. *Annals of Operations Research*, *190*(1), 339-358.

Ai, T. J., & Kachitvichyanukul, V. (2009). A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, *36*(5), 1693-1702.

Akkerman, R., Farahani, P., & Grunow, M. (2010). Quality, safety and sustainability in food distribution: a review of quantitative operations management approaches and challenges. *OR Spectrum*, *32*(4), 863-904.

Alshamrani, A., Mathur, K., & Ballou, R. H. (2007). Reverse logistics: simultaneous design of delivery routes and returns strategies. *Computers & Operations Research*, *34*(2), 595-619.

Alvarenga, G. B., Mateus, G. R., & De Tomi, G. (2007). A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows. *Computers & Operations Research*, *34*(6), 1561-1584.

Amorim, P., Antunes, C. H., & Almada-Lobo, B. (2011). Multi-objective lot-sizing and scheduling dealing with perishability issues. *Industrial & Engineering Chemistry Research*, *50*(6), 3371-3381.

Amorim, P., Günther, H. O., & Almada-Lobo, B. (2012). Multi-objective integrated production and distribution planning of perishable products. *International Journal of Production Economics*, *138*(1), 89-101.

Amorim, P., Meyr, H., Almeder, C., & Almada-Lobo, B. (2013). Managing perishability in production-distribution planning: a discussion and review. *Flexible Services and Manufacturing Journal*, *25*(3), 389-413.

Amorim, P., & Almada-Lobo, B. (2014). The impact of food perishability issues in the vehicle routing problem. *Computers & Industrial Engineering*, *67*, 223-233.

Angelelli, E., & Mansini, R. (2002). The vehicle routing problem with time windows and simultaneous pick-up and delivery. In *Quantitative approaches to distribution logistics and supply chain management* (pp. 249-267). Springer, Berlin, Heidelberg.

Baker, B. M., & Ayechew, M. A. (2003). A genetic algorithm for the vehicle routing problem. *Computers & Operations Research*, *30*(5), 787-800.

Bektas, T. (2006). The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega*, *34*(3), 209-219.

Belenguer, J. M., Benavent, E., & Martínez, M. C. (2005). RutaRep: a computer package to design dispatching routes in the meat industry. *Journal of Food Engineering*, *70*(3), 435-445.

Bell, J. E., & McMullen, P. R. (2004). Ant colony optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics*, *18*(1), 41-48.

Berger, J., & Barkaoui, M. (2003). A hybrid genetic algorithm for the capacitated vehicle routing problem. In *Genetic and Evolutionary Computation Conference* (pp. 646-656). Springer, Berlin, Heidelberg.

Berger, J., & Barkaoui, M. (2004). A parallel hybrid genetic algorithm for the vehicle routing problem with time windows. *Computers & Operations Research*, *31*(12), 2037-2053.

Bianchessi, N., & Righini, G. (2007). Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery. *Computers & Operations Research*, *34*(2), 578-594.

Bogataj, M., Bogataj, L., & Vodopivec, R. (2005). Stability of perishable goods in cold logistic chains. *International Journal of Production Economics*, *93*, 345-356.

Bräysy, O., & Gendreau, M. (2005a). Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. *Transportation Science*, *39*(1), 104-118.

Bräysy, O., & Gendreau, M. (2005b). Vehicle routing problem with time windows, Part II: Metaheuristics. *Transportation Science*, *39*(1), 119-139.

Bullnheimer, B., Hartl, R. F., & Strauss, C. (1999). Applying the ant system to the vehicle routing problem. In *Meta-heuristics* (pp. 285-296). Springer, Boston, MA.

Cao, E., & Lai, M. (2007a). An improved differential evolution algorithm for the vehicle routing problem with simultaneous delivery and pick-up service. In *Third International Conference on Natural Computation (ICNC 2007)* (Vol. 3, pp. 436-440). IEEE.

Cao, E., & Lai, M. (2007b). Vehicle routing problem with simultaneous delivery and pick-up with time windows. In *International Conference on Transportation Engineering* (pp. 160-166).

Carter, C. R., & Rogers, D. S. (2008). A framework of sustainable supply chain management: moving toward new theory. *International Journal of Physical Distribution & Logistics Management*, *38*(5), 360-387.

Chaabane, A., Ramudhin, A., & Paquet, M. (2012). Design of sustainable supply chains under the emission trading scheme. *International Journal of Production Economics*, *135*(1), 37-49.

Chapman, P. A. (2010). Reducing product losses in the food supply chain. In *Delivering Performance in Food Supply Chains* (pp. 225-242). Woodhead Publishing.

Chen, H. K., Hsueh, C. F., & Chang, M. S. (2009). Production scheduling and vehicle routing with time windows for perishable food products. *Computers & Operations Research*, *36*(7), 2311-2319.

Chen, J. F., & Wu, T. H. (2006). Vehicle routing problem with simultaneous deliveries and pickups. *Journal of the Operational Research Society*, *57*(5), 579-587.

Cheng, C. B., & Wang, K. P. (2009). Solving a vehicle routing problem with time windows by a decomposition technique and a genetic algorithm. *Expert Systems with Applications*, *36*(4), 7758-7763.

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

Chu, F., Labadi, N., & Prins, C. (2005). Heuristics for the periodic capacitated arc routing problem. *Journal of Intelligent Manufacturing*, *16*(2), 243-251.

Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, *12*(4), 568-581.

Cook, W., & Rich, J. L. (1999). *A parallel cutting-plane algorithm for the vehicle routing problem with time windows*. Working Paper, Computational and Applied Mathematics, Rice University.

Cordeau, J. F., Gendreau, M., & Laporte, G. (1997). A tabu search heuristic for periodic and multi- depot vehicle routing problems. *Networks: An International Journal*, *30*(2), 105-119.

Cordeau, J. F., Desaulniers, G., Gesrosiers, J., Solomon, M. M., & Soumis, F. (2001a). The VRP with Time Windows. Chapter 7, Paolo Toth and Daniel Vigo (eds), SIAM. *Monographs on Discrete Mathematics and Applications*.

Cordeau, J. F., Laporte, G., & Mercier, A. (2001b). A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society*, *52*(8), 928-936.

Cordeau, J. F., Gendreau, M., Hertz, A., Laporte, G., & Sormany, J. S. (2005). New heuristics for the vehicle routing problem. In *Logistics Systems: Design and Optimization* (pp. 279-297). Springer, Boston, MA.

Çatay, B. (2010). A new saving-based ant algorithm for the vehicle routing problem with simultaneous pickup and delivery. *Expert Systems with Applications*, *37*(10), 6809-6817.

Daganzo, C. F. (1987). Modeling distribution problems with time windows: Part I. *Transportation Science*, *21*(3), 171-179.

Daganzo, C. F. (1987). Modeling distribution problems with time windows. Part II: Two customer types. *Transportation Science*, *21*(3), 180-187.

Dalgaard, P., Buch, P., & Silberg, S. (2002). Seafood Spoilage Predictor—development and distribution of a product specific application software. *International Journal of Food Microbiology*, *73*(2-3), 343-349.

Dantzig, G., Fulkerson, R., & Johnson, S. (1954). Solution of a large-scale traveling-salesman problem. *Journal of the Operations Research Society of America*, *2*(4), 393-410.

Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, *6*(1), 80-91.

De Oliveira, H. C. B., & Vasconcelos, G. C. (2010). A hybrid search method for the vehicle routing problem with time windows. *Annals of Operations Research*, *180*(1), 125-144.

Dell'Amico, M., Righini, G., & Salani, M. (2006). A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection. *Transportation Science*, *40*(2), 235-247.

Desrochers, M., & Soumis, F. (1988). A generalized permanent labelling algorithm for the shortest path problem with time windows. *INFOR: Information Systems and Operational Research*, *26*(3), 191-212.

Desrochers, M., Desrosiers, J., & Solomon, M. (1992). A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research*, *40*(2), 342-354.

Desrosiers, J., Dumas, Y., Solomon, M. M., & Soumis, F. (1995). Time constrained routing and scheduling. *Handbooks in Operations Research and Management Science*, *8*, 35-139.

Dethloff, J. (2001). Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up. *OR-Spektrum*, *23*(1), 79-96.

Dethloff, J. (2002). Relation between vehicle routing problems: an insertion heuristic for the vehicle routing problem with simultaneous delivery and pick-up applied to the vehicle routing problem with backhauls. *Journal of the Operational Research Society*, *53*(1), 115-118.

Donselaar, Van, K. H., & Broekmeulen, R. A. (2012). Approximations for the relative outdating of perishable products by combining stochastic modeling, simulation and regression modeling. *International Journal of Production Economics*, *140*(2), 660-669.

Dror, M., & Trudeau, P. (1990). Split delivery routing. *Naval Research Logistics*, *37*(3), 383-402.

Dror, M., Laporte, G., & Trudeau, P. (1994). Vehicle routing with split deliveries. *Discrete Applied Mathematics*, *50*(3), 239-254.

Ekşioğlu, S. D., & Jin, M. (2006). Cross-facility production and transportation planning problem with perishable inventory. In *International Conference on Computational Science and Its Applications* (pp. 708-717). Springer, Berlin, Heidelberg.

Elhedhli, S., & Merrick, R. (2012). Green supply chain network design to reduce carbon emissions. *Transportation Research Part D: Transport and Environment*, *17*(5), 370-379.

Elkington, J. (1998). Partnerships from cannibals with forks: The triple bottom line of 21st- century business. *Environmental Quality Management*, *8*(1), 37-51.

Faulin, J. (2003). Applying MIXALG procedure in a routing problem to optimize food product delivery. *Omega*, *31*(5), 387-395.

Figliozzi, M. A. (2008). Planning approximations to the average length of vehicle routing problems with varying customer demands and routing constraints. *Transportation Research Record*, *2089*(1), 1-8.

Figliozzi, M. A. (2010). An iterative route construction and improvement algorithm for the vehicle routing problem with soft time windows. *Transportation Research Part C: Emerging Technologies*, *18*(5), 668-679.

Fisher, M. L. (1994). Optimal solution of vehicle routing problems using minimum k-trees. *Operations Research*, *42*(4), 626-642.

Fisher, M. (1995). Vehicle routing. *Handbooks in Operations Research and Management Science*, *8*, 1-33.

Fu, B., & Labuza, T. P. (1993). Shelf-life prediction: theory and application. *Food Control*, *4*(3), 125-133.

Gajpal, Y., & Abad, P. (2009). An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup. *Computers & Operations Research*, *36*(12), 3215-3223.

Gan, X., Wang, Y., Li, S., & Niu, B. (2012). Vehicle routing problem with time windows and simultaneous delivery and pick-up service based on MCPSO. *Mathematical Problems in Engineering*, *2012*.

Ganesh, K., & Narendran, T. T. (2008). TASTE: a two-phase heuristic to solve a routing problem with simultaneous delivery and pick-up. *The International Journal of Advanced Manufacturing Technology*, *37*(11-12), 1221-1231.

Gavish, B., & Graves, S. C. (1978). *The travelling salesman problem and related problems*. Operations Research Center Working Paper OR 078-78. MIT OR Center.

Gélinas, S., Desrochers, M., Desrosiers, J., & Solomon, M. M. (1995). A new branching strategy for time constrained routing problems with application to backhauling. *Annals of Operations Research*, *61*(1), 91-109.

Gendreau, M., Hertz, A., & Laporte, G. (1994). A tabu search heuristic for the vehicle routing problem. *Management Science*, *40*(10), 1276-1290.

Gold, S., Seuring, S., & Beske, P. (2010). The constructs of sustainable supply chain management–a content analysis based on published case studies. *Progress in Industrial Ecology, an International Journal*, *7*(2), 114-137.

Golden, B. L., Magnanti, T. L., & Nguyen, H. Q. (1975). *Implementing vehicle routing algorithms* (No. TR-115). MIT Cambridge OR Center.

Golden, B., Assad, A., Levy, L., & Gheysens, F. (1984). The fleet size and mix vehicle routing problem. *Computers & Operations Research*, *11*(1), 49-66.

Golden, B. L., & Assad, A. A. (1986). OR forum—perspectives on vehicle routing: exciting new developments. *Operations Research*, *34*(5), 803-810.

Golden, B. L., Wasil, E. A., Kelly, J. P., & Chao, I. M. (1998). The impact of metaheuristics on solving the vehicle routing problem: algorithms, problem sets, and computational results. In *Fleet Management and Logistics* (pp. 33-56). Springer, Boston, MA.

Gopalakrishnan, K., Yusuf, Y. Y., Musa, A., Abubakar, T., & Ambursa, H. M. (2012). Sustainable supply chain management: A case study of British Aerospace (BAe) Systems. *International Journal of Production Economics*, *140*(1), 193-203.

Goyal, S. K., & Giri, B. C. (2001). Recent trends in modeling of deteriorating inventory. *European Journal of Operational Research*, *134*(1), 1-16.

Hassini, E., Surti, C., & Searcy, C. (2012). A literature review and a case study of sustainable supply chains with a focus on metrics. *International Journal of Production Economics*, *140*(1), 69-82.

Holland, J.H. (1975). Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence; *MIT Press: Cambridge, MA, USA.*

Hsu, C. I., Hung, S. F., & Li, H. C. (2007). Vehicle routing problem with time-windows for perishable food delivery. *Journal of Food Engineering*, *80*(2), 465-475.

Hwang, H. S. (1999). A food distribution model for famine relief. *Computers & Industrial Engineering*, *37*(1-2), 335-338.

Jiang, W., Zhang, Y., & Xie, J. (2009). A particle swarm optimization algorithm with crossover for vehicle routing problem with time windows. In *2009 IEEE Symposium on Computational Intelligence in Scheduling* (pp. 103-106). IEEE.

Kallehauge, B., Larsen, J., Madsen, O. B., & Solomon, M. M. (2005). Vehicle routing problem with time windows. In *Column Generation* (pp. 67-98). Springer, Boston, MA.

Kallehauge, B. (2008). Formulations and exact algorithms for the vehicle routing problem with time windows. *Computers & Operations Research*, *35*(7), 2307-2330.

Kassem, S., & Chen, M. (2013). Solving reverse logistics vehicle routing problems with time windows. *The International Journal of Advanced Manufacturing Technology*, *68*(1-4), 57-68.

Kim, H., Yang, J., & Lee, K. D. (2009). Vehicle routing in reverse logistics for recycling end-of-life consumer electronic goods in South Korea. *Transportation Research Part D: Transport and Environment*, *14*(5), 291-299.

Kim, H., Yang, J., & Lee, K. D. (2011). Reverse logistics using a multi-depot VRP approach for recycling end-of-life consumer electronic products in South Korea. *International Journal of Sustainable Transportation*, *5*(5), 289-318.

Koç, Ç., Laporte, G., & Tükenmez, İ. (2020). A Review on Vehicle Routing with Simultaneous Pickup and Delivery. *Computers & Operations Research*, *122*, 104987.

Kolen, A. W., Rinnooy Kan, A. H. G., & Trienekens, H. W. (1987). Vehicle routing with time windows. *Operations Research*, *35*(2), 266-273.

Koskosidis, Y. A., Powell, W. B., & Solomon, M. M. (1992). An optimization-based heuristic for vehicle routing and scheduling with soft time window constraints. *Transportation Science*, *26*(2), 69-85.

Labuza, T. P. (1982). Shelf-life dating of foods. *Food & Nutrition Press*, Inc.

Laporte, G., & Nobert, Y. (1987). Exact algorithms for the vehicle routing problem. In *North-Holland Mathematics Studies* (Vol. 132, pp. 147-184). North-Holland.

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

Laporte, G., Gendreau, M., Potvin, J. Y., & Semet, F. (2000). Classical and modern heuristics for the vehicle routing problem. *International Transactions in Operational Research*, *7*(4-5), 285-300.

Larsen, J. (1999). *Parallelization of the vehicle routing problem with time windows*. Lyngby, Denmark: Institute of Mathematical Modelling, Technical University of Denmark.

Lenstra, J. K., & Rinnooy Kan, A. H. G. (1981). Complexity of vehicle routing and scheduling problems. *Networks*, 11(2), 221–227.

Levin, A. (1971). Scheduling and fleet routing models for transportation systems. *Transportation Science*, *5*(3), 232-255.

Li, H., & Lim, A. (2003). Local search with annealing-like restarts to solve the VRPTW. *European Journal of Operational Research*, *150*(1), 115-127.

Lin, S. W., Ying, K. C., Lee, Z. J., & Chen, H. S. (2006, October). Vehicle routing problems with time windows using simulated annealing. In *2006 IEEE International Conference on Systems, Man and Cybernetics* (Vol. 1, pp. 645-650). IEEE.

Lin, C., Choy, K. L., Ho, G. T., Chung, S. H., & Lam, H. Y. (2014). Survey of green vehicle routing problem: past and future trends. *Expert Systems with Applications*, *41*(4), 1118-1138.

Linton, J. D., Klassen, R., & Jayaraman, V. (2007). Sustainable supply chains: An introduction. *Journal of Operations Management*, *25*(6), 1075-1082.

Lukasse, L. J. S., & Polderdijk, J. J. (2003). Predictive modelling of post-harvest quality evolution in perishables, applied to mushrooms. *Journal of Food Engineering*, *59*(2-3), 191-198.

Lütke Entrup, M., Günther, H. O., Van Beek, P., Grunow, M., & Seiler, T. (2005). Mixed-Integer Linear Programming approaches to shelf-life-integrated planning and scheduling in yoghurt production. *International Journal of Production Research*, *43*(23), 5071-5100.

Magnanti, T. L. (1981). Combinatorial optimization and vehicle fleet planning: Perspectives and prospects. *Networks*, *11*(2), 179-213.

Manzini, R., & Accorsi, R. (2013). The new conceptual framework for food supply chain assessment. *Journal of Food Engineering*, *115*(2), 251-263.

Mao, H., Qu, F., & Li, X. (2007). Integration of forward and reverse logistics at the vehicle routing level. In *International Conference on Transportation Engineering 2007* (pp. 3518-3523).

Matai, R., Singh, S., & Mittal, M. L. (2010). Traveling salesman problem: an overview of applications, formulations, and solution approaches. In *Traveling Salesman Problem, Theory and Applications*. IntechOpen.

McDonald, K., & Sun, D. W. (1999). Predictive food microbiology for the meat industry: a review. *International Journal of Food Microbiology*, *52*(1-2), 1-27.

Mentzer, J. T., DeWitt, W., Keebler, J. S., Min, S., Nix, N. W., Smith, C. D., & Zacharia, Z. G. (2001). Defining supply chain management. *Journal of Business Logistics*, *22*(2), 1-25.

Min, H. (1989). The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transportation Research Part A: General*, *23*(5), 377-386.

Min, H., & Zhou, G. (2002). Supply chain modeling: past, present and future. *Computers & Industrial Engineering*, *43*(1-2), 231-249.

Mingyong, L., & Erbao, C. (2010). An improved differential evolution algorithm for vehicle routing problem with simultaneous pickups and deliveries and time windows. *Engineering Applications of Artificial Intelligence*, *23*(2), 188-195.

Mohan, S., Gopalakrishnan, M., & Mizzi, P. J. (2013). Improving the efficiency of a non-profit supply chain for the food insecure. *International Journal of Production Economics*, *143*(2), 248-255.

Mole, R. H., & Jameson, S. R. (1976). A sequential route-building algorithm employing a generalised savings criterion. *Journal of the Operational Research Society*, *27*(2), 503-511.

Montanari, R. (2008). Cold chain tracking: a managerial perspective. *Trends in Food Science & Technology*, *19*(8), 425-431.

Montané, F. A. T., & Galvao, R. D. (2006). A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Computers & Operations Research*, *33*(3), 595-619.

Mosheiov, G. (1994). The travelling salesman problem with pick-up and delivery. *European Journal of Operational Research*, *79*(2), 299-310.

Mosheiov, G. (1998). Vehicle routing with pick-up and delivery: tour-partitioning heuristics. *Computers & Industrial Engineering*, *34*(3), 669-684.

Myers, D. C. (1997). Meeting seasonal demand for products with limited shelf lives. *Naval Research Logistics*, *44*(5), 473-483.

Nagy, G., & Salhi, S. (2005). Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *European Journal of Operational Research*, *162*(1), 126-141.

Nahmias, S. (1982). Perishable inventory theory: A review. *Operations Research*, *30*(4), 680-708.

Naso, D., Surico, M., Turchiano, B., & Kaymak, U. (2007). Genetic algorithms for supply-chain scheduling: A case study in the distribution of ready-mixed concrete. *European Journal of Operational Research*, *177*(3), 2069-2099.

O'Connor, A. D., & De Wald, C. A. (1970). A sequential deletion algorithm for the design of optimal transportation networks 37th national meeting of the Operations Research Society of America. *Bulletin of the Operations Research Society of America*, *18*(1).

Osman, I. H. (1993). Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, *41*(4), 421-451.

Osvald, A., & Stirn, L. Z. (2008). A vehicle routing algorithm for the distribution of fresh vegetables and similar perishable food. *Journal of Food Engineering*, *85*(2), 285-295.

Pagell, M., & Wu, Z. (2009). Building a more complete theory of sustainable supply chain management using case studies of 10 exemplars. *Journal of Supply Chain Management*, *45*(2), 37-56.

Parragh, S. N., Doerner, K. F., & Hartl, R. F. (2008a). A survey on pickup and delivery models part ii: Transportation between pickup and delivery locations. *Journal für Betriebswirtschaft*, *58*(2), 81-117.

Parragh, S. N., Doerner, K. F., & Hartl, R. F. (2008b). A survey on pickup and delivery problems. *Journal für Betriebswirtschaft*, *58*(1), 21-51.

Pawsey, R. K. (1995). Preventing losses and preserving quality in food cargoes. *Food, Nutrition and Agriculture-15-Food safety and trade*. Food and Agriculture Organization (FAO) of the United Nations.

Pillac, V., Gendreau, M., Guéret, C., & Medaglia, A. L. (2013). A review of dynamic vehicle routing problems. *European Journal of Operational Research*, *225*(1), 1-11.

Potvin, J. Y., Kervahut, T., Garcia, B. L., & Rousseau, J. M. (1996). The vehicle routing problem with time windows part I: tabu search. *INFORMS Journal on Computing*, *8*(2), 158-164.

Potvin, J. Y., & Bengio, S. (1996). The vehicle routing problem with time windows part II: genetic search. *INFORMS Journal on Computing*, *8*(2), 165-172.

Privé, J., Renaud, J., Boctor, F., & Laporte, G. (2006). Solving a vehicle-routing problem arising in soft-drink distribution. *Journal of the Operational Research Society*, *57*(9), 1045-1052.

Prindezis, N., Kiranoudis, C. T., & Marinos-Kouris, D. (2003). A business-to-business fleet management service provider for central food market enterprises. *Journal of Food Engineering*, *60*(2), 203-210.

Prins, C. (2004). A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, *31*(12), 1985-2002.

Qi, M. Y., Miao, L. X., Zhang, L., & Xu, H. Y. (2008, September). A new tabu search heuristic algorithm for the vehicle routing problem with time windows. In *2008 International Conference on Management Science and Engineering 15th Annual Conference Proceedings* (pp. 1648-1653). IEEE.

Qureshi, A. G., Taniguchi, E., & Yamada, T. (2009). An exact solution approach for vehicle routing and scheduling problems with soft time windows. *Transportation Research Part E: Logistics and Transportation Review*, *45*(6), 960-977.

Qureshi, A. G., Taniguchi, E., & Yamada, T. (2010). Exact solution for the vehicle routing problem with semi soft time windows and its application. *Procedia-Social and Behavioral Sciences*, *2*(3), 5931-5943.

Qureshi, A. G., Taniguchi, E., & Yamada, T. (2012). A microsimulation based analysis of exact solution of dynamic vehicle routing with soft time windows. *Procedia-Social and Behavioral Sciences*, *39*, 205-216.

Raafat, F. (1991). Survey of literature on continuously deteriorating inventory models. *Journal of the Operational Research Society*, *42*(1), 27-37.

Reimann, M., Doerner, K., & Hartl, R. F. (2004). D-ants: Savings based ants divide and conquer the vehicle routing problem. *Computers & Operations Research*, *31*(4), 563-591.

Renaud, J., Laporte, G., & Boctor, F. F. (1996). A tabu search heuristic for the multi-depot vehicle routing problem. *Computers & Operations Research*, *23*(3), 229-235.

Rong, A., Akkerman, R., & Grunow, M. (2011). An optimization approach for managing fresh food quality throughout the supply chain. *International Journal of Production Economics*, *131*(1), 421-429.

Ropke, S., Cordeau, J. F., & Laporte, G. (2007). Models and branch- and- cut algorithms for pickup and delivery problems with time windows. *Networks*, *49*(4), 258-272.

Rusdiansyah, A., & Tsao, D. B. (2005). An integrated model of the periodic delivery problems for vending-machine supply chains. *Journal of Food Engineering*, *70*(3), 421-434.

Russell, R. A. (1977). An effective heuristic for the m-tour traveling salesman problem with some side conditions. *Operations Research*, *25*(3), 517-524.

Salhi, S., & Nagy, G. (1999). A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. *Journal of the Operational Research Society*, *50*(10), 1034-1042.

Savelsbergh, M. W. (1985). Local search in routing problems with time windows. *Annals of Operations Research*, *4*(1), 285-305.

Schrage, L. (1981). Formulation and structure of more complex/realistic routing and scheduling problems. *Networks*, *11*(2), 229-232.

Schultmann, F., Zumkeller, M., & Rentz, O. (2006). Modeling reverse logistic tasks within closed-loop supply chains: An example from the automotive industry. *European Journal of Operational Research*, *171*(3), 1033-1050.

Seuring, S., & Müller, M. (2008). From a literature review to a conceptual framework for sustainable supply chain management. *Journal of Cleaner Production*, *16*(15), 1699-1710.

Sexton, T. R., & Choi, Y. M. (1986). Pickup and delivery of partial loads with "soft" time windows. *American Journal of Mathematical and Management Sciences*, *6*(3-4), 369-398.

Shi, Y., Zhou, Y., Boudouh, T., & Grunder, O. (2020). A lexicographic-based two-stage algorithm for vehicle routing problem with simultaneous pickup–delivery and time window. *Engineering Applications of Artificial Intelligence*, *95*, 103901.

Sloof, M., Tijskens, L. M. M., & Wilkinson, E. C. (1996). Concepts for modelling the quality of perishable products. *Trends in Food Science & Technology*, *7*(5), 165-171.

Smith, D., & Sparks, L. (2004). Temperature controlled supply chains. *Bourlakis, MA and PWH Weightman, Food Supply Chain Management*, 179-198.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, *35*(2), 254-265.

Solomon, M. M., & Desrosiers, J. (1988). Survey paper—time window constrained routing and scheduling problems. *Transportation Science*, *22*(1), 1-13.

Srivastava, S. K. (2007). Green supply chain management: a state-of-the-art literature review. *International Journal of Management Reviews*, *9*(1), 53-80.

Stricker, R. (1970). *Public sector vehicle routing: the Chinese Postman Problem* (Doctoral dissertation, MIT).

Subramanian, A., Drummond, L. M. D. A., Bentes, C., Ochi, L. S., & Farias, R. (2010). A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, *37*(11), 1899-1911.

Subramanian, A., Uchoa, E., Pessoa, A. A., & Ochi, L. S. (2011). Branch-and-cut with lazy separation for the vehicle routing problem with simultaneous pickup and delivery. *Operations Research Letters*, *39*(5), 338-341.

Taillard, É., Badeau, P., Gendreau, M., Guertin, F., & Potvin, J. Y. (1997). A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science*, *31*(2), 170-186.

Tan, K. C., Lee, L. H., Zhu, Q. L., & Ou, K. (2001). Heuristic methods for vehicle routing problem with time windows. *Artificial Intelligence in Engineering, 15*(3), 281-295.

Tang, J., Pan, Z., Fung, R. Y., & Lau, H. (2009). Vehicle routing problem with fuzzy time windows. *Fuzzy Sets and Systems*, *160*(5), 683-695.

Tarantilis, C. D., & Kiranoudis, C. T. (2001). A meta-heuristic algorithm for the efficient distribution of perishable foods. *Journal of Food Engineering*, *50*(1), 1-9.

Tarantilis, C. D., & Kiranoudis, C. T. (2002). Distribution of fresh meat. *Journal of Food Engineering*, *51*(1), 85-91.

Tasan, A. S., & Gen, M. (2012). A genetic algorithm based approach to vehicle routing problem with simultaneous pick-up and deliveries. *Computers & Industrial Engineering*, *62*(3), 755-761.

Thangiah, S. R. (1993). *Vehicle routing with time windows using genetic algorithms*. Artificial Intelligence Lab., Slippery Rock University.

Thangiah, S. R., Potvin, J. Y., & Sun, T. (1996). Heuristic approaches to vehicle routing with backhauls and time windows. *Computers & Operations Research*, *23*(11), 1043-1057.

Thangiah, S. R., & Petrovic, P. (1998). Introduction to genetic heuristics and vehicle routing problems with complex constraints. In *Advances in computational and stochastic optimization, logic programming, and heuristic search* (pp. 253-286). Springer, Boston, MA.

Thangiah, S. R., & Salhi, S. (2001). Genetic clustering: an adaptive heuristic for the multidepot vehicle routing problem. *Applied Artificial Intelligence*, *15*(4), 361-383.

Toth, P., & Vigo, D. (2003). The granular tabu search and its application to the vehicle-routing problem. *INFORMS Journal on Computing*, *15*(4), 333-346.

Toth, P., & Vigo, D. (Eds.). (2014). Vehicle routing: problems, methods, and applications. *MOS-SIAM Series on Optimization.*

Trienekens, J., & Zuurbier, P. (2008). Quality and safety standards in the food industry, developments and challenges. *International Journal of Production Economics*, *113*(1), 107-122.

Van der Vorst, J. G., & Beulens, A. J. (1999). A research model for the redesign of food supply chains. *International Journal of Logistics: Research and Applications*, *2*(2), 161-174.

Van der Vorst, J. G., Beulens, A. J., & van Beek, P. (2000). Modelling and simulating multi-echelon food systems. *European Journal of Operational Research*, *122*(2), 354-366.

Van der Vorst, J. G., van Kooten, O., Marcelis, W. J., Luning, P. A., & Beulens, A. J. (2007). Quality controlled logistics in food supply chain networks: integrated decision-making on quality and logistics to meet advanced customer demands. *14th International Annual Euroma Conference*, Ankara.

Van der Vorst, J. G., Tromp, S. O., & Zee, D. J. V. D. (2009). Simulation modelling for food supply chain redesign; integrated decision making on product quality, sustainability and logistics. *International Journal of Production Research*, *47*(23), 6611-6631.

Van Donk Pieter, D., Akkerman, R., & Van der Vaart, T. (2008). Opportunities and realities of supply chain integration: the case of food manufacturers. *British Food Journal*, *110*(2), 218-235.

Vellema, S., & Boselie, D. M. (2003). *Cooperation and competence in global food chains: perspectives on food quality and safety*. Shaker.

Verbič, M. (2004). *Econometric estimation of parameters of preservation of perishable goods in cold logistic chains*. Institute for Economic Research.

Vlajic, J. V., Van der Vorst, J. G., & Haijema, R. (2012). A framework for designing robust food supply chains. *International Journal of Production Economics*, *137*(1), 176-189.

Wang, Y. (2008). Study on the model and tabu search algorithm for delivery and pickup vehicle routing problem with time windows. In *2008 IEEE International Conference on Service Operations and Logistics, and Informatics* (Vol. 1, pp. 1464-1469). IEEE.

Wang, W., Wang, Z., & Qiao, F. (2008). An improved genetic algorithm for vehicle routing problem with time-window. In *2008 International Symposium on computer science and computational technology* (Vol. 1, pp. 189-194). IEEE.

Wang, X., & Li, D. (2012). A dynamic product quality evaluation based pricing model for perishable food supply chains. *Omega*, *40*(6), 906-917.

Wassan, N. A., Wassan, A. H., & Nagy, G. (2008). A reactive tabu search algorithm for the vehicle routing problem with simultaneous pickups and deliveries. *Journal of Combinatorial Optimization*, *15*(4), 368-386.

Wittstruck, D., & Teuteberg, F. (2012). Understanding the success factors of sustainable supply chain management: empirical evidence from the electrics and electronics industry. *Corporate Social Responsibility and Environmental Management*, *19*(3), 141-158.

Wognum, P. N., Bremmers, H., Trienekens, J. H., Van der Vorst, J. G., & Bloemhof, J. M. (2011). Systems for sustainability and transparency of food supply chains–Current status and challenges. *Advanced Engineering Informatics*, *25*(1), 65-76.

Yan, C., Banerjee, A., & Yang, L. (2011). An integrated production–distribution model for a deteriorating inventory item. *International Journal of Production Economics*, *133*(1), 228-232.

Zachariadis, E. E., & Kiranoudis, C. T. (2011). A local search metaheuristic algorithm for the vehicle routing problem with simultaneous pick-ups and deliveries. *Expert Systems with Applications*, *38*(3), 2717-2726.

Zanoni, S., & Zavanella, L. (2007). Single-vendor single-buyer with integrated transport-inventory system: Models and heuristics in the case of perishable goods. *Computers & Industrial Engineering*, *52*(1), 107-123.

Zanoni, S., & Zavanella, L. (2012). Chilled or frozen? Decision strategies for sustainable food supply chains. *International Journal of Production Economics*, *140*(2), 731-736.

Zhang, G., Habenicht, W., & Spieß, W. E. L. (2003). Improving the structure of deep frozen and chilled food chain with tabu search procedure. *Journal of Food Engineering*, *60*(1), 67-79.

Zhen, F. (2011). Multi-period vehicle routing problem with recurring dynamic time windows. In *ICSSSM11* (1-6). IEEE.

Zhen, L., Lv, W., Wang, K., Ma, C., & Xu, Z. (2020). Consistent vehicle routing problem with simultaneous distribution and collection. *Journal of the Operational Research Society*, *71*(5), 813-830.

Ziggers, G. W., & Trienekens, J. (1997). Quality Assurance in Food and Agribusiness Supply Chains: Developing successful partnerships. *International Journal of Production Research*, *60-61*, 271-279.

# APPENDICES

**A.** Test problems section 5.2

https://drive.google.com/file/d/1BAhJxDLDIdYOVh-VSTpmfVl1dvwsMA6T/view?usp=sharing

**B.** Test problems section 5.3.1

https://drive.google.com/file/d/1KlWCxi5TNSfVIiiOPlZaglPPn-WL2HUe/view?usp=sharing

**C.** Test problems section 5.3.2

https://drive.google.com/file/d/1uso44kZfd8qOrpMaQRoft4zXQWaFMyTB/view?usp=sharing

**D.** Test problems section 5.3.3

https://drive.google.com/file/d/1uLbpK8syOWdB-mUBhdes0FfNMr9RmpZf/view?usp=sharing

**E.** Table 5.21- Sensitivity q%

https://drive.google.com/file/d/1EcgvHyvMZAfJdoaZ5CpPQENgatQo2nHn/view?usp=sharing

**F.** Table 5.22- Sensitivity TW%

https://drive.google.com/file/d/1M5MgbLroXopqTXam-3-wgamGvjkfjwMJ/view?usp=sharing

**G.** Table 5.23- Sensitivity ab%

https://drive.google.com/file/d/1QqHAX-5YAmk0TDpOH4DnzXPCkNP2lrt6/view?usp=sharing

## CURRICULUM VITAE

### PERSONAL INFORMATION

Surname, Name: Farshchi, Mahdi
Nationality: Iranian
Date and Place of Birth: 19 September 1975, Tabriz
Marital Status: Single
Phone: +90 536 9221441
e-mail: meh.farshchi@gmail.com

### EDUCATION

| Degree | Institution | Year of Graduation |
|---|---|---|
| MS | University of Azad, South Tehran, Industrial Engineering | 2003 |
| BS | Iran University of Science and Technology Industrial Engineering | 1999 |
| High School | Ferdoosi High School, Tabriz | 1994 |

### WORK EXPERIENCES

| Year | Place | Enrollment |
|---|---|---|
| 2005-2011 | Tabriz Azad University<br>Department of Management, Accounting and Economy | Faculty member |

### FOREIGN LANGUAGES

Advanced English, Turkish

### PUBLICATIONS

1. Farshchi M, Paydarfar D, Industrial safety- Safety Systems and Human Factors Engineering, (2009), Iran.
2. Farshchi M, Hoshvand M, Operations Research, (2010), Iran.

### HOBBIES

Football, Body Building, Computer Technologies, Movies, Industrial Design