The Vehicle Routing Problem in Omni-Channel Retail Distribution Systems

by

Mohamed Mahmoud Saleh Abdulkader

A Thesis submitted to The Faculty of Graduate Studies of

The University of Manitoba

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Mechanical Engineering

University of Manitoba

Winnipeg

August 2017

Abstract

The emergence of huge industries and mega economies has led to the creation of supply chains and the development of transportation systems. Supply chain management is used to direct raw materials and products through the supply chains, which were extended beyond the direct relation between the producer and customer. The vehicle routing problem (VRP) is concerned with one of the operational decisions in supply chain. The VRP is the class of problems in which the demand of customers should be fulfilled by a fleet of vehicles. This fleet of vehicles is starting from and returning to the depot while minimizing the total route cost such as travelled distance, time, etc. The VRP is considered a more sophisticated form of the famous TSP.

In this thesis, VRP models arising in omni-channel retail distribution systems are introduced. Retail distribution systems are considered as omni-channel systems when consumers can either place orders online or visit the stores physically to buy the products. In case of online orders, the organizations are responsible for the products delivery. The resultant VRP models can be considered as new variants of the VRP. These models represent variety of scenarios adopted by different retail chain store organizations.

Mathematical formulations are provided for these new variants of the VRP and solved to obtain optimum solutions for small problem instances. The VRP and its variants are NP-hard problems and difficult to solve in the case of large problem instances. Therefore, different heuristics and metaheuristics are proposed to obtain optimum (or near optimum) solutions for large problem instances. New bench mark problem instances are generated to test the proposed heuristics and metaheuristics performance. The average percentage deviation of the proposed metaheuristics with respect to the optimum solutions of small problem instances obtained from the mathematical models is less than 1%. It can be concluded that the proposed metaheuristics have good performance while succeeding to keep shorter calculation time.

Acknowledgements

In the name of ALLAH, the Entirely Merciful, the Especially Merciful. All praise and Gratitude is due to ALLAH, the Lord of the worlds.

I would like to express my deepest gratitude to my advisor Prof. Tarek ElMekkawy for his continues support, inspiration, motivation, and direction throughout my journey of seeking this degree. Special thanks go to my co-advisor Dr. Yuvraj Gajpal. His encouragement, guidance, support and time were essential to complete this work. I would like to thank my committee members Prof. Qingjin Peng and Prof. Srimantoorao S. Appadoo and Prof. Satyaveer S. Chauhan for their valuable comments that assisted in enhancing this work.

I thank my colleagues and my friends for their support and motivation. I thank the staff at the Department of Mechanical Engineering and the Faculty of Graduate Studies. I acknowledge the financial support from the NSERC and the University of Manitoba to complete this work.

I would like to express my sincere gratitude to the elementary basis in my life; my beloved family. My parents, I only reached this moment because of you. My loving sister and best friend, thanks for your continuous support. My lovely wife, Mayada, I guess you earned this degree. Finally, my precious daughters, I overcome the hardships of life for you.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AC	Ant Colony
CVRP	Capacitated Vehicle Routing Problem
GA	Genetic Algorithm
HAC	Hybridized Ant Colony
MAC	Multi Ant Colony
MCVRP	Multi-Compartment Vehicle Routing Problem
MIP	Mixed Integer Programming
OR	Operations research
PD	Percentage Deviation
PDP	Pickup and Delivery Problem
PDTSP	Pickup and Delivery Travelling Salesman Problem
PSO	Particle Swarm Optimization
RPD	Relative Percentage Deviation
SA	Simulated Annealing
TS	Tabu Search
TSP	Travelling Salesman Problem
VRP	Vehicle Routing Problem
VRPSPD	Vehicle Routing Problem with Simultaneous Pickup and Delivery
VRPTW	Vehicle Routing Problem with Time Windows

Chapter 1

Introduction

1.1 Background

Transportation of goods through supply chains has grabbed vast interest in the optimization research through last decades. Transportation has been a key element in the competition between organizations as it can denote 10% to 20% or more of the cost of any product [1]. Organizations have realized that the struggle in markets is not determined only by the product quality anymore. The market is controlled by many other factors, i.e. how far the organizations can spread their products and how much they can satisfy the customers' needs at the lowest cost.

With the rise of the global awareness towards the environmental concerns, many legislations have been achieved regarding the waste management. In 1991, a law has been established in Germany which obligates each industry to manage its own packaging waste. The industry must bear the cost of collecting, sorting, and recycling waste if it did not recall back that waste [2]. This increased the transportation cost of firms and the complexity of the transportation systems. The attention in optimizing the transportation process increased even more because it is one of the main consumers of fossil fuel which causes CO_2 emissions.

Operations research (OR) has been employed in transportation planning. It was proven to be very efficient in optimizing the distribution of goods. The VRP is the problem of distributing goods between depots and customers. In this class of problems, the demands of customers are fulfilled with the products originating from depot and transported using a fleet of vehicles in such a way that the total travelling cost of all vehicles is minimized. The VRP was first considered as a generalized form of the famous Travelling Salesman Problem (TSP). It was formulated by Dantzig and Ramser [3] to find shortest route to deliver fuel using gasoline delivery trucks from central depot to gas stations. There are many forms of the VRP according to the different classifications of customers and vehicles characteristics.

The vehicle characteristics:

- The vehicles may be assumed with unlimited capacity and can fulfill any amount of demand or the problem can be capacitated.
- There might be restrictions on the distance and/or time for the vehicles trips.
- The vehicle can be identical or heterogeneous.
- The vehicles can be composed of one or multi number of compartments.
- The number of the vehicle in the fleet can be constrained or not.
- The vehicles can be settled in single or multiple depots.

- Travel time can be static, dynamic, or stochastic.

The customers' characteristics:

- Customers can restrict the start and the end of delivery time.
- All customers may require the same or different services (demand and pickup).
- Customers' orders can be transportations requests to other customers.
- Customers may require similar or different type of goods.
- Demand can be known or real time demand.
- Demand can be satisfied in one visit or multiple visits (i.e. split demand).

1.2 Motivation and Objectives

According to the Global Powers of Retailing report [4], the 250 largest retailers around the world generated revenues of US$4.31 trillion in 2015. This figure indicates that despite the challenging global economy in 2015, retailers managed to keep steady growth and international expansion. The revenue generated by e-commerce was 8.7 percent of the total retail revenue of the top 250 retailing companies. The volume of e-commerce sales has been steadily growing in recent years. Companies working in retail industry are motivated to adopt different strategies and policies to cope up with this market change. The increasing share of e-commerce in retail industry clearly exhibits the shift of retail industry from single channel to omni-channels [5]. Omni-channel retailing is one of the recently developed business models. In this model, customers can order products online and get these products delivered at their homes or visit the store physically to get the products. The delivery of the online ordered products to the customers is the responsibility of the company. To increase the customer satisfaction, companies are working on making all the products available in stores as well as online. In addition, they are trying to reduce the time between orders placement and delivery times. This means that online orders should be satisfied from nearby retail stores. The companies usually operate their own distribution networks to deliver products to different retail stores in a certain city. Each distribution network includes a central warehouse, and a group of scattered retail stores. A fleet of vehicles is used to deliver products from the center warehouse to the retail stores. This fleet of vehicles can be utilized to deliver online orders from retail stores to consumers as well. This research is motivated from the supply chain management of omni-channel retailing distribution systems. Different scenarios are investigated to provide applicable solutions for the optimization of omni-channel retailing distribution networks.

1.3 Contribution

In this work, different scenarios in retail industry distribution systems are investigated. Each of the investigated scenarios represents a new variant of the VRP with real life application in retail industry. I provide a description for these new variants and develop mathematical formulations. The mathematical formulations are solved to obtain optimum solutions for small size problem instances. In addition, I present different approximate solution approaches which can obtain optimum (or near optimum) solutions. Finally, I introduce new bench mark problems to test the performance of the proposed solution approaches and to test other algorithms in future work. The main contributions of this research work can be summarized as follows:

- **The multi-compartment VRP**

I provide a mathematical formulation for the multi-compartment VRP (MCVRP). I propose a new hybridized Ant Colony (AC) algorithm to solve the problem. I perform computational experiments on new generated bench mark problem instances. The new algorithm improved the existing solutions for almost all problem instances. Finally, I investigate the benefit of using multi-compartment vehicles instead of the regular vehicle. The multi compartment feature will be used later in one of the proposed VRP models for omni-channel retailing distribution systems.

- **The TSP for online ordered products in retail industry**

I introduce a new variant of the TSP that arises in the delivery of online ordered products in retail industry. We investigate this problem as it can be considered as a simplified VRP with only one vehicle route. Exploring this model provides a better understanding of the VRP model in omni-channel retail distribution systems. I provide a description of the problem and present its mathematical formulation to obtain optimal solutions for the problem in small problem instances. In addition, I propose a heuristics and a metaheuristic to obtain near optimum solutions for the problem in large problem instances. I present new bench mark problem instances to test the proposed algorithms. I compare the proposed algorithms against the optimal solution obtained by solving the mathematical formulation.

- **The MCVRP for omni-channel retail distribution**

I introduce the VRP arising in omni-channel retail distribution. I describe the problem and provide a mathematical formulation to solve the problem optimally. In this problem, the whole distribution network in omni-channel retailing is investigated. It considers the distribution of the products from the warehouse to the retail stores and delivery of online orders from the retail stores to the consumers. The model suggests utilizing the same fleet of vehicles in both types of transportation by using multi-compartment vehicles in satisfying the demand of the retail stores and the online consumers simultaneously. I propose different heuristics and a metaheuristic to solve the problem. I generate new bench mark problem instances to evaluate the proposed algorithms and test algorithms in future work.

- **The VRP in omni-channel retailing distribution systems**

I introduce a new variant of the VRP arising in omni-channel retailing distribution systems. The new problem can be considered a generalization of both capacitated VRP and the pickup and delivery problem (PDP). This model considers the assignment of consumers to retail stores as a decision variable along with the routing problem. Consumers are assigned to retail stores to satisfy the orders of the consumers based on the product availability at the retail stores inventory. This decision can be made simultaneously while selecting the best routes. The model is different from the previous models because the previous models (3-4) assume that the consumers are already assigned to retail stores and the assignment decisions are known in advance. In addition, the model adds more complexity to the previous model by assuming maximum route length which is more realistic. I provide a description for the problem and explain the generalized VRP structure. I present a mathematical formulation to solve the problem to optimality. I propose a two-phase heuristic and multi-AC algorithm to solve large problem instances. I generate new bench mark problem instances to evaluate the performance of the proposed solution approaches. Finally, I illustrate the benefit of using the proposed generalized VRP structure instead of the two existing structure in case of omni-channel distribution systems.

1.4 Thesis Organization

In chapter 2, a review of the related literature is presented. In chapter 3, the multi-compartment VRP is studied and the benefit of using multi compartments is investigated. Chapter 4 introduces the TSP for the delivery of online ordered products in retail industry. Chapter 5 introduces the multi- compartment VRP in omni-channel retail distribution. Chapter 6 introduces the VRP arising in omni-channel retailing distribution systems. Chapter 7 concludes the thesis and presents the future work directions.

Chapter 2

Literature Review

For simplification the VRP will be classified according to the service required by customers. All customers may require only one service type (demand or pickup). However, some of them may require one service while others require the other services. Moreover, in some cases customers may require both service types. A summary of the literature review with extensive classification can be found in the next subsection.

2.1 Customers Require One Service Type

Lorini et al. [6] investigated the dynamic VRP to minimize the sum of travel time and lateness at the customers and depot. They allowed redirection of vehicles to other locations instead of planned destinations after communication between the dispatch office and the drivers. These communications is used to update the drivers with new customer requests and the dispatch office about current drivers' locations. They assumed a fleet of K identical incapacitated vehicles dispatched from single depot and soft time windows.

Lin [7] investigated the benefit of coordination between non- identical resources, in minimizing the total cost. Light resources are used to deliver items or transported with a heavier resource (which can deliver items itself). They formulated the problem using MIP model and presented two stages heuristic to solve it. They assumed incapacitated problem but limited capacity for heavier resource in term of the light resources. They assumed that resources can perform multiple routes and pickup time window for customers.

2.1.1 Capacitated VRP

In the capacitated VRP (CVRP) a fleet of identical vehicles (initially settled in the depot) serves a number of customers. Each vehicle visits a group of the customers only once in a predetermined order called route, such that the total demand of the customers does not exceed the capacity of the vehicle. A more constrained form is VRP with time windows (VRPTW) in which the deliveries to the customers are restricted by start and end times. Opening and closing times may be considered to the depot as well.

The CVRP was first considered a generalized form of the TSP and formulated by Dantzig and Ramser [3]. In the last six decades, many exact, heuristic, and meta-heuristic approaches have been proposed to solve the problem. A branch-and-cut algorithm based on a two-commodity network flow formulation was presented by Baldacci et al. [8]. An exact algorithm based on set partitioning formulation was described by Fukasawa et al. [9]. New formulations for the problem and new lower bounds were presented by Letchford and Salazar-Gonzalez [10]. Different types of metaheuristics have been efficiently used to solve the CVRP (e.g., hill climber heuristic (Derigs and Kaiser [11]), particle swarm optimization (PSO) (Ai and Kachitvichyanukul [12]), memetic algorithm (Nagata and

Bräysy [13]), artificial bee colony algorithm (Szeto et al. [14]), and AC algorithm (Reimann et al. [15], and Yu et al. [16])). Toklu et al. [17] studied the CVRP with time window constraints under travel time uncertainty. They proposed an AC algorithm to minimize the costs. Brandao [18] proposed a Tabu Search (TS) to minimize the total cost in heterogenous fixed fleet vehicle routing problem HFFVRP. They used fixed fleet of different capacity vehicles to satisfy the customers demand. For recent research in VRP and its variants refer to (Cordeau et al. [19]; Golden et al. [20]; Laporte [21]; Eksioglu et al. [22]; Toth and Vigo [1]; Lahyani et al. [23]).

2.1.2 Multiple Routes

The VRP with multiple routes is an extension of the basic VRP when vehicles are required to make more than one trip per day due to limitation in the time or the number of available vehicles and/or drivers. Azi et al. [24] formulated the problem of VRPTW using set portioning and introduced a branch and cut algorithm. They used a fleet of K identical vehicles which have limited capacity Q and perform multiple routes. Because the limited capacity they tried to choose which customers to be served based on difference between their revenue and traveling costs.

Macedo et al. [25] proposed a set partitioning formulation and exact algorithm based on pseudo-polynomial network flow model for the VRPTW to minimize the total travelled distance and increase the number of served customers. A fleet of K identical vehicles with capacity Q can make multiple trips.

Cattaruzza et al. [26] considered the multi trip VRP in which a fleet of K identical vehicles with limited capacity Q is used to serve the customers demand. They proposed a hybrid Genetic Algorithm (GA) to minimize the total travel time. They advised considering time window constrains in the future work.

2.1.3 Multi Commodity

Reed et al. [27] considered the CVRP for the waste collection. They assumed a fleet of identical multi compartment vehicles with capacity Q to transfer the separated waste. They minimized the total route length using AC system algorithm. They suggested the extension of the simple two-compartment model to more compartments.

Cattaruzza et al. [28] proposed an iterated local search algorithm to reduce the number of used vehicles and minimize the cost in a multi commodity multi trip VRPTW. They used a fleet of identical vehicles with capacity Q and that commodities are incompatible and cannot be transferred together. They recommended that multi trip should be used to increase the utilization of the vehicles.

2.1.4 Split Deliveries

The split delivery VRP is the relaxation of CVRP in which the customer can be visited more than once. This relaxation can yield to total distance minimization as well as the number of vehicles. Moreover, it can be used in extending the VRP to a planning horizon with periodic demands.

Salani and Vacca [29] presented a flow-based Mixed Integer Programming (MIP) for the split delivery VRPTW and proposed a branch and price algorithm to minimize the total travel time. They assumed delivery dependent service times and discrete instead of continuous split deliveries which is more realistic. They used a fleet of K identical vehicles with capacity Q.

Belfiore and Yoshizaki [30] presented a mathematical formulation and developed a scatter- search approach to minimize the sum of the fixed vehicle costs and the travel costs in the split delivery VRPTW. They assumed unlimited fleet of heterogeneous vehicles with different capacities. They suggested using a scatter-search approach in the multi depot VRP and the PDP in the future.

2.1.5 Multi Depot

Increased demand and scattered markets required more distribution channels. The multi depot VRP requires more decisions to be made; which depot customers will be served by. Obviously, all depots are identical and offer the same commodity.

Escobar et al. [31] presented a hybrid Granular TS algorithm to minimize the total traveling costs in the multi depot VRP. They assumed identical fleet of K vehicles with capacity Q and a maximum trip time T. They suggested using heterogeneous fleet of vehicles in future work.

Contardo and Martinelli [32] presented a vehicle-flow and a set-partitioning formulation to minimize the total traveling time in the multi depot VRP. They used cutting planes to solve the first formulation and column-and-cut generation to solve the second. They used identical fleet of K vehicles with capacity Q and a maximum trip time T. They suggested extending their model to multi depot VRPTW, multiple-echelon VRP or multiple-period VRP.

Gulczynski et al. [33] investigated the effect of split deliveries on the total distance reduction in MDVRP. They formulated a MIP for the problem and presented a heuristic to maximize the total distance saving. They assumed capacitated problem. They recommended considering minimum delivery amounts on vehicles when splitting and using time windows as future directions.

Salhi et al. [34] presented a flow based MIP formulation and a variable neighborhood search meta-heuristic. They addressed the problem of minimizing the total cost and determining the composition of the fleet in multi depot VRP. They used unlimited fleet of heterogeneous vehicles with different capacities and maximum route time. They forced the vehicles to return to their original depots, this constrain relaxation can be investigated as future direction and may lead to cost reduction.

2.2 Customers Require Demand and Pickup Services

With the introduction of reversed and closed-loop supply chains; another flow of goods in the form of pickups from customers to depots originated. To increase vehicle utilization three different strategies are suggested to deal with the combined pickups and deliveries. The first strategy is to schedule the pickups after all the deliveries are satisfied in the vehicle route back to the depot in VRP with backhauls. In the second strategy, VRP with Mixed Backhauls, mixed orders of the linehauls and backhauls customers are integrated.

2.2.1 Mixed Backhauls

Dondo and Cerda [35] presented a MIP formulation and a neighborhood search algorithm to solve the mixed backhauls VRPTW. They assumed soft time windows and maximum trip duration. They assumed a fleet of heterogeneous vehicles with different capacities. They intended to work on the dynamic environment VRP.

2.2.2 Simultaneous Pickup and Delivery

The third strategy allows the customers to have deliveries and pickups which are done simultaneously in one visit. This problem is known by VRP with Simultaneous Pickup and Delivery (VRPSPD). All strategies can be constrained with time windows.

Gajpal and Abad [36] presented an AC system with three local search procedures: 2-opt, insertion/interchange, and arc exchange. Their algorithm can solve the VRP with backhaul and mixed load as well. Gajpal and Abad [37] proposed a saving heuristic and a parallel saving heuristic. They checked the feasibility of merging two routes using cumulative net pickup approach. Subramanian et al. [38] addressed the VRPSPD and proposed a MIP model based on undirected and directed two-commodity flow formulations. They presented a branch and cut algorithm to solve the problem trying to minimize the total routing cost. Zachariadis et al. [39] presented an adaptive memory algorithm to reduce the total cost of a route in VRPSPD. They used a central depot where a fleet of homogeneous K vehicles with capacity Q is used to serve the customers.

Goksal et al. [40] considered the VRPSPD and proposed a hybridization approach of the PSO and variable neighborhood descent algorithm. They used unlimited fleet of identical vehicles with capacity Q and tried to minimize the total routing cost. They suggested extending the model to solve the multi depot VRPSPD.

Wang and Chen [41] introduced the flexible delivery and pickup problem with time windows which improves the mixed backhauls VRPTW by allowing simultaneous pickup and delivery to reduce the cost. They presented a MIP model to minimize the number of used vehicles and the total traveled distance. They proposed a co-evolutionary algorithm based on GA to solve the problem. They assumed distribution and collection centers. They suggested extending their model to allow demand uncertainty.

Liu et al. [42] introduced a special generalized form of VRPSPD with time windows in which other transportations services are provided to the customers; delivery from hospital to patient and pickup from patient to medical lab. The problem can be considered multi commodity generalization. They proposed two MIP formulations to the problem and used GA and TS to minimize the total routing cost.

2.2.3 The Pickup and Delivery Problem

In VRPSPD the pickups are directed to the depot and prohibited to other customers. The transportation orders between customers are the distinguishing feature of the PDP. However, each customer has either pickup or delivery only.

The PDP was considered for the first time by Lokin [43]. The author introduced a variant of the TSP where precedence relations are forced on some of the customers. This means that some nodes must be visited before other nodes. The author described a branch and bound algorithm for this problem, which was later known as the PDP. Kalantari et al. [44] proposed a branch and bound algorithm for the problem. They considered single and multiple capacitated and non-capacitated vehicles. Savelsbergh and Sol [45] presented a survey and a description of the general pickup and delivery problem. Lu and Dessouky [46] provided MIP formulation for the multiple vehicle PDP and proposed a branch-and-cut algorithm. Ting and Liao [47] formulated the selective PDP and presented a memetic algorithm to solve the problem.

The PDP with time windows is a generalization of the PDP. Ropke and Cordeau [48] formulated the problem of PDP with time windows using set partitioning and introduced a branch-and-cut-and-price algorithm to solve the problem. They used identical vehicles with capacity Q located in a center depot and tried to minimize the total routing cost. Baldacci et al. [49] presented a new algorithm for the same problem using set partitioning and solved it using branch-and-cut-and-price also. Their objective function was to minimize the sum of vehicle fixed and route costs. They reported that their proposed algorithm is faster and solved previously unsolved instances.

Different heuristics and metaheuristics have been used to solve the problem (e.g., a reactive TS (Nanry and Barnes [50]), a two-phase heuristic (Lau and Liang [51]), a tabu-embedded simulated annealing (SA) (Li and Lim [52]), a grouping GA (Pankratz [53]), a large neighborhood search heuristic (Ropke and Pisinger [54]), an insertion-based construction heuristic (Lu and Dessouky [55]), a SA, a PSO, a GA, and an artificial immune system (D'Souza et al. [56])).

The PDP with transfers is an extension of the PDP with time windows in which the pickup request can reach its final destination by another vehicle. The transfer between vehicles occurs at certain locations usually called transfer points where the drivers may be switched as well. Masson et al. [57] introduced the PDP with Shuttle routes which is a special case of the PDP with transfer. They presented Arc-based and Set Partitioning formulations and proposed a branch-and-cut-and-price algorithm to solve the problem optimally to minimize the total distance traveled by the vehicle fleet. They assumed time windows and homogeneous fleet of vehicles K with capacity Q stationed in a single depot. They suggested considering a heterogeneous fleet of vehicles.

Rais et al. [58] allowed transshipment in the network as well. They formulated the problem with and without time window constrains using MIP model. They assumed flexible size fleet of heterogeneous vehicles with different capacities. They used a commercial solver that uses a combination of branch-and-cut and branch-and-bound techniques to minimize the total cost of the vehicle routes.

Rieck et al. [59] considered the many-to-many location problem with inter-hub transport with multi commodity. They did not assume time windows and used identical vehicles. They considered the facility positioning to minimize the facility fixed cost, facility operating costs, and routing costs. A MIP was formulated, and a fix-and-optimize scheme procedure and a GA were proposed. A number of survey papers that deal with the PDP have been recently published. Berbeglia et al. [60] presented a survey and classification for the problem. Parragh et al. [61-62] presented a comprehensive survey and another classification.

2.3 Solution Approaches

Since the introduction of the TSP and the VRP to the operations research literature, many exact, heuristic, and meta-heuristic approaches have been proposed to solve the different variants of the VRP. Here, I present some of the most commonly used metaheuristics in solving different variants of the VRP.

The AC algorithm is inspired by the behavior of ants in the search of food. They mark their trails by laying a substance called pheromone. The amount of laid pheromone on the path inspires other ants to know whether this path is promising or not. This observation inspired Dorigo et al. [63] to design a metaheuristic technique to solve combinatorial optimization problems. They presented the first AC to solve the TSP in which agents called ants simulate the behavior of the real ants. Due to simplicity of the general procedures of the algorithm, it was applied in many different problems.

The AC has been used in solving the VRP and its variants since Bullnheimer et al. [64] designed their AC trying to solve the basic VRP. It was used to solve: the CVRP (Reimann et al. [15], Yu et al. [16], Reed et al. [27], Bin et al. [65]), the VRPTW (Toklu et al. [17]), MCVRP (Reed et al. [27], and Abdulkader et al. [66]), the VRPSPD (Gajpal and Abad [36]), the dynamic VRP (Montemanni et al. [67]), the time dependent VRPTW (Balseiro et al. [68]), the VRPTW with heterogeneous vehicles and multiple products (De la Cruz et al. [69]).

The GA was initially proposed and developed by J. Holland [70] in the 1960s and 1970s. It was inspired by the evolution process and its natural mechanisms; mating and mutation. Several extensions to these algorithms have been developed and for several years genetic algorithms have been efficient in solving different combinatorial optimization problems. The GA has been used in solving different variants of the VRP since it was first applied to solve the TSP by Brady [71]. It was used to solve: the VRPTW (Berger and Barkaoui [72]), the multi depot VRP (Ho et al. [73]), the multi trip VRP (Cattaruzza et al. [26]), the periodic VRPTW (Nguyen et al [74]), the VRPSPD (Wang and Chen [41] and Liu et al. [42]), and the PDP (Pankratz [53], D'Souza et al. [56], and Rieck et al. [59]).

The PSO was inspired by the social behaviour of bird flocking and fish schooling by Kennedy and R. Eberhart [75]. The PSO has been used in solving different variants of the VRP: the CVRP (Ai and Kachitvichyanukul [12]), the multi depot VRP (Geetha et al. [76]), the heterogeneous VRP (Yao et al. [77]), the VRP with stochastic demand (Marinakis et al. [78]), the VRPSPD (Goksal et al. [40]), and the PDP (D’Souza et al. [56]).

The SA was inspired by the physical annealing process in which the temperature of the material is reduced by slow cooling rate. The algorithm is using random search to decrease the objective function. The SA was first used to solve the TSP by Kirkpatrick et al. [79]. Since then it has been used in solving different variants of the VRP: the CVRP (Osman [80]), the multi depot VRP (Mirabi et al. [81]), the VRPTW (Chiang and Russell [82]), the VRPSPD (Wang et al. [83]), and the PDP (Li and Lim [52] and D’Souza et al. [56]).

The TS was presented by Glover [84] to find better near optimum solutions for combinatorial problems. Its idea is to examine neighbour solutions and select the best neighbour. The selected neighbours are banned from being selected in next iterations by adding them to a tabu list to prevent trapping in cycles of improvement. One of the first trials to apply TS for solving the VRP was done by Willard [85]. The TS has been used in solving different variants of the VRP: the CVRP (Osman [80]), the CVRP with heterogenous vehicles (Brandao [18]), the multi depot VRP (Escobar et al. [31]), the VRPTW (Badeau et al. [86]), the VRPSPD (Liu et al. [42]), and the PDP (Nanry and Barnes [50]).

2.4 Conclusions

From the previous review of the VRP and its variants, it can be concluded that different solution approaches have been proposed to solve different scenarios for the delivery of goods. However, the rise of e-commerce motivated the organization that operates in retail industries to adopt a recently developed business models which is called omni-channel retailing. In this model, customers can order products online and get these products delivered at their home or by physically visiting the store. It allows consumers to go shopping seven days a week, twenty-four hours a day. The delivery of the online ordered products to the customers is the responsibility of the company. In addition, the companies that own chain stores operate their own distribution networks to deliver the products from the warehouse to different retail stores located in the city. This work considers the issue of managing these complicated distribution systems of the omni-channel retail business model. The VRP models arising in omni-channel distribution system have not been yet considered in the literature. This thesis will be the first one to introduce VRP models arising in omni-channel distribution system. There are different VRP structures available in the OR literature. However, none of these structures exactly fits in to the structure of distribution systems arising in omni-channels. The VRP structures arising in omni-channel distribution systems are complex although they have similarity with some of the VRP structures available in literature. In This thesis, I propose different VRP models arising in omni-channel distribution systems. These models will provide practical solutions for the supply chain management in retail industry.

Chapter 3

Multi-Compartment Vehicle Routing Problem

3.1 Introduction

The MCVRP deals with satisfying the demand of customers with different products. The demand of each customer for each product is constant and known in advance. The products should be stored in different compartments of the same vehicle while being transported together. Vehicles are partitioned into a constant number of compartments with certain capacities. Customers are assigned to routes so that the total demand of customers assigned to any route for certain product does not exceed the capacity of the reserved compartment for this product. The objective is to minimize the total transportation cost.

Another application of the MCVRP is the delivery of food and grocery where refrigerated and non-refrigerated grocery items are stored in two different compartments in the same vehicle. Chajakis and Guignard [87] presented two integer programming models for two different layout vehicles and presented Lagrangean Relaxations for the first model. They addressed the decisions involved in assigning customers to routes only and stated that sequencing of customers within trips can be solved by any TSP algorithm. They stated that the customers' orders should be fulfilled completely by one vehicle.

Another application of the MCVRP is delivering different types of fuels using a fleet of vehicles or marine vessels with different capacity tanks. To solve this problem, Avella et al. [88] used set portioning to formulate a branch and bound algorithm. El Fallahi et al. [89] found an application in which animal foods is supplied to the farms separately. They proposed two algorithms to solve the problem; memetic algorithm and TS algorithm.

Muyldermans and Pang [90] investigated the benefits of co-collection of sorted waste from different locations to central location by multi compartment vehicles over separate collection by regular vehicles. They introduced new local search procedure based on 2-opt, cross, exchange, and relocate moves to solve the problem and compared their results with El Fallahi et al. [89]. It was assumed in [88-90] that the demand of each customer for certain items cannot be splitted. However, they assumed that more than one vehicle can visit the customer to fulfil demands of different items. Reed et al. [27] proposed an AC with 2-opt local search improvement to solve the basic CVRP in recycling waste collection network. They extended their algorithm to solve the MCVRP in which the customers are visited only once by one vehicle. Our work is inspired from Reed et al. [27] algorithm extension for the MCVRP. I address the problem of food and grocery delivery from retail store to different customer locations where two types of grocery should be stored in different compartments on the same vehicle. I use a fleet of identical vehicles; each one of them visits a group of locations such that each customer is visited by only one vehicle and only once. The problem is to decide which customers are assigned together in one trip as well as the order of visiting them with the objective of minimizing the total travelling distance.

The AC algorithm is inspired by the behavior of ants in the search of food. They mark their trails by laying a substance called pheromone. The amount of laid pheromone on the path inspires other ants to know whether this path is promising or not. This observation inspired Dorigo et al. [63] to design a metaheuristic technique to solve combinatorial optimization problems. They presented the first AC in which agents called ants simulate the behavior of the real ants. The ants communicate with each other by the pheromone laid by the ants while travelling from one place to another. Higher amount of pheromone in a path increases the probability of ants to follow that path. Dorigo et al. [63] used the TSP to apply their algorithm and compare it with other approaches. Due to simplicity of the general procedures of the algorithm, it was applied in many different problems.

The AC has been used in solving the VRP and its variants since Bullnheimer et al. [64] designed their AC trying to solve the basic VRP. Although they could not improve the best-known solutions, their algorithm produced good results and showed competiveness with other metaheuristics. Montemanni et al. [67] used the AC to solve the dynamic VRP. They introduced new bench mark problems and tested their algorithm which showed good results. Bin et al. [65] presented an improved AC by offering new pheromone updating strategy called ant-weight and by introducing mutation operator. Other than computation times, their algorithm was effective compared to other metaheuristics.

Gajpal and Abad [36] presented an AC to solve the VRP with simultaneous delivery and pickup. Balseiro et al. [68] tried to enhance the AC algorithm by hybridizing it with an insertion algorithm to solve a time dependent the VRP with time window constraints. De la Cruz et al. [69] proposed a sequential algorithm with AC and TS to solve the VRP with time window constrains in which heterogeneous vehicles deliver multiple products. All these papers presented new best-known solution for the bench mark problems. This shows the effectiveness of the AC algorithm in solving the VRP and its variants.

3.2 Mathematical Modelling

A list of notations related to the problem definition is presented below:

N Set of customers

k Set of vehicles

p Set of compartments and products

q_{ip} Quantity of product *p* to be picked up from customer *i*

Q_p Vehicle capacity reserved for product *p*

L The maximum length of any route

C_{ij} The distance of traversing arc (*ij)*

Q_{ip}^{k} The total carried quantity of product *p* by vehicle *k* after leaving vertex *i*

In this section, a mathematical formulation is provided for the problem. The mathematical formulation for the MCVRP considered in this section is not available in the literature. However, a mathematical formulation for a variant of MCVRP is available in the literature [89]. In the variant of MCVRP, the customer is allowed to get served more than once. Our formulation is based on pickup of materials from customer locations. Let

G= (V, A) be an undirected graph with a set of vertices V= {0, 1 …… n}, where 0 represents the depot node, and N= {1, 2 ….n} are the customers served by a number of identical vehicles k (initially located in the depot). Vehicles are divided into a number of compartments p equals to the number of products handled in the network. Each customer i has a known quantity q_{ip} to be picked up of each product p and each customer is visited exactly once by only one vehicle. Each vehicle visits a group of customers, such that the total load of this group of customers for certain type of products does not exceed the vehicle capacity of the compartment reserved for this product Qp. The maximum length of each route cannot exceed L. Let C_{ij} be the distance of traversing arc (ij). Let X_{ij}^{k} be a binary variable equals to 1 if and only if vehicle k visits customer j just after customer i. Let Q_{ip}^{k} denote the total carried quantity of product p by vehicle k after leaving vertex i. Then, the MCVRP can be formulated as follows:

Minimize

$$Z = \sum_{k\in K}\sum_{i\in V}\sum_{j\in V} C_{ij}X_{ij}^{k} \tag{3.1}$$

Subject to:

$$\sum_{k\in K}\sum_{j\in N} X_{ij}^{k} = 1 \qquad \forall i \in N, i \neq j \tag{3.2}$$

$$\sum_{k\in K}\sum_{i\in N} X_{ij}^{k} = 1 \qquad \forall j \in N, i \neq j \tag{3.3}$$

$$\sum_{i\in N} X_{0i}^{k} = \sum_{j\in N} X_{j0}^{k} = 1 \qquad \forall k \in K \tag{3.4}$$

$$q_{ip} \leq Q_{ip}^{k} \leq Q_p \qquad \forall i \in N, k \in K, p \in P \tag{3.5}$$

$$Q_{ip}^{k} - Q_{jp}^{k} + Q_p X_{ij}^{k} \leq Q_p - q_{jp} \qquad \forall i \in V, j \in V, i \neq j, k \in K, p \in P \tag{3.6}$$

$$\sum_{i\in V}\sum_{j\in V} C_{ij}X_{ij}^{k} \le L \qquad \forall k \in K, i \neq j \tag{3.7}$$

$$X_{ij}^{k} \in \{0,1\} \qquad \forall i \in V, j \in V, i \neq j, k \in K \tag{3.8}$$

Eq. (3.1) is the objective function representing the total cost of all traversed arcs by all vehicles. Eq. (3.2) and (3.3) impose that exactly one arc enters and leaves each vertex associated with a customer respectively. They together ensure that only one vehicle visits a customer and only once. Eq. (3.4) ensures that each vehicle *k* starts its route from depot 0 and ends it at depot 0. Eqs. (3.5) and (3.6) are necessary for sub-tour elimination; they impose the capacity requirement and connectivity requirement between two customers. When $X_{ij}^{k} = 0$, eq. (3.6) is not binding since $Q_{ip}^{k} \le Q_p$ and $q_{jp} \le Q_{jp}^{k}$. When $X_{ij}^{k} = 1$ equations (3.5) and (3.6) impose that $Q_{ip}^{k} - Q_{jp}^{k} \le -q_{jp}$ which eliminates sub-tour construction. In addition, eq. (3.5) ensures that the total carried quantity after visiting vertex *i* does not exceed the vehicle capacity for this product. Thus, eq. (3.5) ensures capacity violation constraint. Eq. (3.7) represents the route length constraint. Eq. (3.8) describes variables X_{ij}^{k} which equal to 1 if and only if vehicle *k* visits customer *j* just after customer *i*.

3.3 Hybridized Ant Colony Algorithm

A list of notations related to the proposed hybridized AC (HAC) is presented below:

τ_{ij}	Trail intensity on arc (*ij)*
m	Number of ants
μ_{ij}	The inverse of the distance of arc (ij)
ε_{ij}	The attractive value of arc (ij)
N_i	The list of all feasible customers that has not been visited by the ant
L^{best}	The length of the best solution found so far
ρ	The trail persistence
$Tabu_s$	The tabu list for first visited customer by all ants.
$Tabu_i$	The tabu list for the customers visited by ant *i*

The AC algorithm is an algorithm used for finding near optimal solutions for NP-hard problems. The algorithm is based upon the behavior of real ants for finding shortest path when they travel from their nest to the food source. In the AC algorithm, artificial ants are used to construct a solution. These simple agents called ants travel from one customer location to another customer location to construct proposed routes (solutions of the VRP). Each ant performs four basic activities during route building process: 1) It chooses next customer based on a probability function of two attraction measures; the distance from current location to the proposed customer and the trail intensity on this arc; 2) It keeps a history of the visited customers during current route (tabu list); 3) It updates the remaining capacity in the vehicle; and 4) It updates trail intensities on the visited arcs. Then local search procedures are applied to enhance the solutions quality. Finally, the tabu lists are deleted and a new iteration starts. The HAC algorithm can be described below and is followed by detailed description in next subsections.

Step 1: Create initial solution to initialize the trail intensities.

Step 2: Repeat the following until termination condition is reached.

- Construct routes for *m* number of ants.
- Perform local search scheme to improve the solution produced by each ant.
- Update best solution found.
- Update trail intensities for all arcs using best solution.

Step 3: Terminate the algorithm and report the best solution.

The flow diagram for generating an initial solution is shown in Figure 3.1. The flow diagram for the proposed HAC is shown in Figure 3.2.

3.3.1 Initial Solution and Initialization of Trail Intensities

Trail intensities are initialized at the beginning of the algorithm. The initial trail intensities do not affect the proceeding of the algorithm because the same amounts are deposited to all edges at the beginning. However, an initial solution is needed to initialize trail intensities. The initial solution is generated randomly by starting the route from the depot. Customers are added randomly to a vehicle route as long as the vehicle capacity is sufficient to pick up the customers load. Otherwise, the vehicle returns to the depot before visiting the next customer. The initial pheromone is calculated based on the expression $\tau_{ij} = 1/L \quad (\forall i, j \in N)$, where L is the total length of a randomly generated tour.

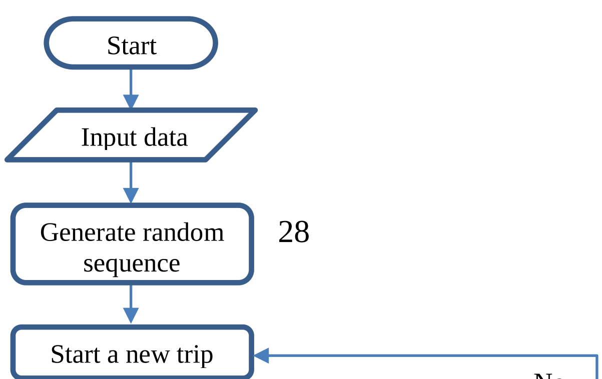

Figure 3.1. The flow diagram of generating initial solution

3.3.2 Route Construction

I use *m* ants to construct ant solutions (see parameters setting in section 4 for setting the value of *m*). Each ant generates a complete tour (a complete solution of MCVRP). Each ant starts its route from the depot and continues by selecting first customer randomly (each ant starts the route from different customer). The first customer is chosen randomly to diversify the solution. Each ant then travels from customer to another customer based on two attraction measures representing the probability function. The first measure is the trail intensity τ_{ij} which contains information on how frequently arc *ij* has been used in previous iterations. The second measure μ_{ij} is the inverse of arc distance which represents the move desirability. The attractive value is calculated as follows:

$$\varepsilon_{ij} = (\tau_{ij})^{\alpha}(\mu_{ij})^{\beta} \tag{3.9}$$

The probability of picking customer j as the next customer is:

$$P_{ij} = \begin{cases} \frac{\varepsilon_{ij}}{\sum_{l \in N_i} \varepsilon_{il}} & if\ j \in N_i \\ 0 & otherwise \end{cases} \tag{3.10}$$

Where N_i is the list of all feasible customers that has not been visited by the current vehicle and can be visited without exceeding the vehicle capacity or passing the maximum length of the trip. However, before each ant decides to choose the next customer according to eq. (3.9), a random variable q uniformly distributed over [0, 1] is assessed. If $q > q_\circ$ the next customer is chosen according to eq. (3.10), otherwise choose the customer which has the maximum attractive value among feasible customers, where $q_\circ$ is a constant value. Each ant continues adding customers to its route until there is no more feasible customers (N_i is empty). Then the ant proceeds to the depot to start another trip.

3.3.3 Local Search Procedures

After creating all routes, three local search procedures; 2-opt, customer insertion, and customer exchange are performed to enhance the solution quality. The local search schemes are implemented sequentially on each route without violating the capacity constraints. The following are the descriptions of the schemes.

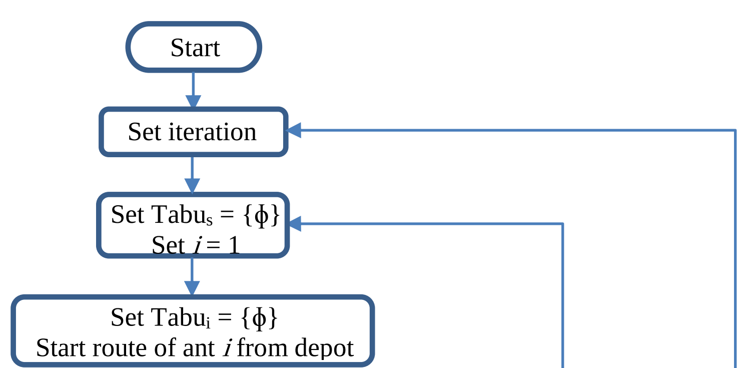

Figure 3.2. The flow diagram of the proposed HAC

3.3.3.1 2-opt Local Search

This scheme was proposed by Lin [91] to solve the TSP. Each trip is considered a TSP, so it can be modified to reduce the distance travelled without exceeding the capacity. A 2-opt local search can be described as follows: The trip is broken at two places forming three sections. The customers' sequence of the middle section is inverted. The trip is reconstructed by connecting the three sections and the total length of the new trip is calculated. All the possible 2-opt moves are investigated by trying all combinations of two places and the move with maximum improvement is chosen. The process continues till no further improvement is possible. It is considered one of the powerful local search schemes for the TSP. However, its implementation in the VRP is limited since customers are not allowed to move between different routes. Thus, I consider two other local search schemes where customers are moved between two routes.

3.3.3.2 Customer Insertion Local Search

The drawback of 2-opt heuristic is that it does not move customers between routes. On the contrary of the 2-opt scheme, customer insertion allows the movement of customers between different trips. There are two types of insertions; insertion in the same trip, and insertion in different trips. Customer i is first removed from position a in trip T_p, then inserted in another position b in T_q. I consider the insertion possibility for customer i in the same trip T_p as well in another trip T_q. If the customer is inserted in the original trip T_p, the capacity constraint will be conserved. Thus, it is only needed to check the possible insertion at different places in the same trip T_p. If insertion is considered in different trip T_q, the capacity of the vehicle is first checked before checking insertion of customer i in trip T_q. If the vehicle capacity permits insertion, maximum route length constraint should be tested for every position. The insertion of customer *i* will be tried in all positions of all trips.

The improvement is calculated by subtracting the increase in length of the enlarged trip due to customer insertion from the reduction in length of the original trip. All customers should be tested as candidates for insertion in all positions. Customer which may yield highest improvement is selected for possible insertion. The current solution is modified by inserting the chosen customer at the place where its insertion yields highest improvement. Once a modified/ improved tour is obtained, the whole process is repeated with modified / improved tour to make a second modified tour. If second modification is not possible, then the search process is stopped. Otherwise, search process is continued till search process stops improving solutions.

3.3.3.3 Customer Swapping Local Search

Customer exchange can be done within the same trip or between different trips. Customer *i* first removed from its position in trip T_p to be swapped with customer *j* in tour T_q. If both customers are in the same trip, then the capacity constraint is not violated. However, if two customers are from different trips, both trips new vehicle loads are checked. In addition, both trip lengths should be checked for maximum length constrains to make sure that the move is feasible. All feasible trades are evaluated and the one with maximum improvement is chosen. The whole swapping process continues on improved tour till no further improvement is possible.

3.3.4 Updating Trail Intensities

After local search is performed, all trail intensities are updated by the best solution found so far. The purpose of the local search schemes is to improve the solutions generated by ants. Therefore, the best solution is updated only after applying local search schemes. Hence, the best solution is then used to update the trail intensities. Pheromone or trail intensities evaporate with time on all arcs. At the same time, ants leave pheromone on the visited arcs. These two actions are performed after local search improvements in each iteration. Updating trail intensities is described in two steps: lowering of pheromone on all arcs and pheromone increase on the arcs reported in the best route only. According to Dorigo et al. [63], ρ is defined as trail persistence $0 \le \rho < 1$. The term (1- ρ) is interpreted as trail evaporation. Eq. (3.11) updates the trail intensities by adding $1/L^{best}$ to the remaining amount of trail intensity after its evaporation.

$$\tau_{ab}^{new} = \begin{cases} \rho \times \tau_{ab}^{old} + 1/L^{best} & if\ arc\ ab \in best\ route \\ \rho \times \tau_{ab}^{old} & otherwise \end{cases} \tag{3.11}$$

Where L^{best} is the total length of the best solution in each iteration. See parameters setting in section 4 for setting the value of ρ. The first part of eq. (3.11) represents the remaining amount of trail intensities after evaporation. Therefore, the existing trail intensities are multiplied by the term ρ. In the second part of eq. (3.11), trail intensities of the arcs included in the best solution are increased by amount $1/L^{best}$.

3.4 Numerical Experiments

In this section, I present the setting of parameters, explain how the data was generated from the bench mark problems, present the numerical results, and compare the results of the proposed (HAC) algorithm with the best reported results of the existing (ACS) proposed by Reed et al. [27]

3.4.1 Parameters Setting

The number of the used ants (m) controls the solution quality, but it also affects the computational time. I found that after 20 ants, the computational time increases excessively without any observed solution improvements. Dorigo et al. [63] found that setting (α=1 and β=2) gives very good solutions. Setting (ρ=0.9) gave a good chance to update the pheromone with the new experience of ants on the account of the existing experiences. In addition, I set (q_o=0.9) as recommended by Reed et al. [27]. I use 100 iterations to test the performance of the HAC against 2000 iterations for the ACS. I use 100 iterations for two reasons: 1) to keep the solution time comparable with the existing algorithm; 2) Improvement is very minimum after 100 iterations.

3.4.2 Data Generation

The new bench mark problems for the MCVRP are generated from the VRP bench mark problems of Christofides [92]. I use the method described by Reed et al. [27] to generate the data set. Christofides [92] has 14 problem data sets named as vrnpc1 to vrnpc14. Each of the original problem was used to generate two different problems referred to as *a* and *b*. For example, problem vrnpc1 was used to generate vrnpc1a and vrnpc1b. The used procedure to generate the two data sets can be described in the following steps:

Step 1: The vehicles were splitted into two compartments using capacity ratio 1:3. This means that if the capacity of the vehicle in the original data set is Q, then the capacity of the two compartments in the new data sets is 0.25Q and 0.75Q.

Step 2: The customers were divided into two regions according to their coordinates; region 1 and region 2. Customers are assigned to their regions as follows: Let X_{max} and

Y_{max} be the maximum x-coordinate and the maximum y-coordinate among all customers coordinates respectively. Region 1 contains the customers with x and y coordinates greater than 0 and less than $X_{max}/2$ and $Y_{max}/2$ respectively. Region 2 contains all other customers, i.e whose x and y coordinates greater than $X_{max}/2$ and $Y_{max}/2$ respectively.

Step 3: The customers located in region 1 require a product pickup quantity of the ratio 1:2. If the pickup quantity in the original data set is q, then the pickup quantity of product 1 and product 2 in the new data sets are $0.33q$ and $0.67q$. in region 1, both problem sets *a* and *b* has the same ratio 1:2 for pickup quantity of product 1 and product 2.

Step 4: Customers from region 2 require a product pickup quantity of the ratio 1:3 in problems vrnpcXa and ratio 1:4 in problems vrnpcXb. If the pickup quantity in the original data set is q, then the pickup quantity of product 1 and product 2 in the new data sets are $0.25q$ and $0.75q$ in problems *a*, and $0.2q$ and $0.8q$ in problems *b*.

In this way I generate 28 problem instances from the 14 problem set of Christofides [92]. The results of the existing ACS are not available for these 28 new generated problem instances. Therefore, I codded the existing ACS to get its solutions for these problem instances.

3.5 Computational Results

The proposed HAC algorithm and the existing ACS were coded in C and the new generated 28 problems were solved. The HAC was run for 100 iterations only to maintain the considerably low computational time. The ACS was run for 2000 iterations as recommended by Reed et al. [27]. The CPU times reported in Table 3.1 and Table 3.2 are in seconds. The results were obtained using a server that operates four 2.1GHz processors with 16-core each and a total of 256 GB of RAM. The results are presented in Table 3.1. The percentage improvement in total length is calculated according to eq. (3.12) by dividing the difference in average total length calculated by both algorithms by the total length calculated by ACS.

$$Improvement\ \% = \frac{ACS - HAC}{ACS} * 100 \tag{3.12}$$

The results reported in Table 3.1 show that the average total length of ACS is 1109.2 unit length and the average computational time is 204.5 seconds, the average total length of HAC is 1048.4 unit length and the average computational time is 128.5 seconds. The average total length improvement of the proposed HAC over the existing ACS is 5.1%. These results already exhibit that the proposed algorithm produces better quality solution in less computational time compared to the existing algorithm for all solved problems except one problem (vrpnc12a). This validates the dominance of the proposed algorithm over the existing algorithm. The difference between the total length calculated by HAC and ACS is very tight in problems vrnpc11 to vrnpc14. This is because the customers are clustered in these problems and thus the problems can be solved easily. A trend can be remarked in the total length improvement percentage with the number of customers. It increases from about 3% in small problems to more than 8% in larger problems. This trend indicates that the performance of the proposed HAC over the existing ACS increases with the increase in problem size. It can be concluded that HAC maintains its high performance in large problems where the number of customers is more than 100. Note that the ACS algorithm used 2000 iterations while the proposed HAC used 100 iterations. The HAC used less number of iterations but produced better solutions because of the hybridization of the AC algorithm with local search procedures. The effect of using different hybridization schemes is further analyzed and described in the next section.

Table 3.1: The results of the generated problems

Problem	No. of customers	ACS		HAC		Improvement %
		Total Length	CPU Time (s)	Total Length	CPU Time (s)	
vrpnc1a	50	569.564	16	550.70	5	3.31
vrpnc1b	50	569.118	17	551.94	5	3.02
vrpnc2a	75	957.525	36	890.68	15	6.98
vrpnc2b	75	954.856	35	918.96	14	3.76
vrpnc3a	100	964.132	122	874.07	40	9.34
vrpnc3b	100	959.327	122	895.26	44	6.68
vrpnc4a	150	1253.86	345	1126.12	146	10.19
vrpnc4b	150	1254.51	336	1159.48	151	7.58
vrpnc5a	199	1587.02	688	1444.29	257	8.99
vrpnc5b	199	1640.59	676	1525.87	236	6.99
vrpnc6a	50	573.274	13	557.49	11	2.75
vrpnc6b	50	573.378	13	559.37	10	2.44
vrpnc7a	75	997.007	33	928.24	28	6.90

vrpnc7b	75	969.337	32	932.67	26	3.78
vrpnc8a	100	963.381	97	882.96	93	8.35
vrpnc8b	100	976.212	97	884.85	95	9.36
vrpnc9a	150	1343.08	273	1228.88	326	8.50
vrpnc9b	150	1346.63	274	1226.58	333	8.91
vrpnc10a	199	1645.58	606	1511.65	624	8.14
vrpnc10b	199	1659.94	608	1526.02	620	8.07
vrpnc11a	120	1133.88	281	1110.45	75	2.07
vrpnc11b	120	1247.49	280	1221.73	87	2.06
vrpnc12a	100	911.861	105	912.64	15	-0.09
vrpnc12b	100	970.833	100	950.79	30	2.06
vrpnc13a	120	1577.45	171	1556.46	117	1.33
vrpnc13b	120	1572.11	168	1550.12	123	1.40
vrpnc14a	100	914.857	91	911.35	34	0.38
vrpnc14b	100	970.933	91	965.84	38	0.52
Average		1109.205	204.5	1048.41	128.50	5.14

3.5.1 Effect of Hybridization

In this section, I present the effect of hybridizing the AC algorithm with the local search schemes. In order to observe the effect of local search schemes on HAC algorithm, I solve the MCVRP under two settings. In the first setting, the HAC algorithm described in section 3 is used without adding any local search schemes to solve the problem. In the second setting, the HAC is combined with local search schemes under three criteria. Under the first criteria, only 2-opt local search scheme is combined with HAC. Under the second criteria, 2-opt and swapping local search schemes are used. Finally, under the third criteria, 2-opt, swapping and insertion local search schemes are all combined with HAC. I change the number of iterations for each criterion to maintain approximately the same computational time. I evaluate the performance of the algorithm after applying each local search scheme. The results are shown in Table 3.2. Figure 3.3 represents the effect of hybridization on the solution quality. It can be noticed that hybridizing the AC algorithm with local search schemes enhanced the solution quality. When the AC is hybridized, improvement percentage in the average total distance is calculated from the average total length obtained by AC algorithm when no local search scheme is applied.

It is clear from Table 3.2 and Figure 3.3 that the AC algorithm need better local search schemes for better performance. When 2-opt local search is combined with AC, the solution quality is improved by 11.31%. However, 2-opt local search is not sufficient to improve the solution quality. Note that 2-opt local search is applied only on a single trip and thus its performance is good for the TSP but its performance for the VRP is limited. When the AC is combined with 2-opt and swapping local search schemes, the solution quality improves from 11.31% to 12.66%. The results indicate that swapping enhances the solution quality because it moves customers between two routes. Finally, when insertion local search is combined with 2-opt and swapping, the solution improves from 12.66% to 17.11%. This result shows that insertion local search is very effective local search scheme in improving the solution quality.

Table 3.2: The effect of hybridization on solution quality

Degree of Hybridization	Number of Iterations	Average Total Distance	CPU Time (s)	Improvement %
No local search	10000	1264.78	141.82	0.00
2-opt	10000	1121.70	127.96	11.31
2-opt + Swapping	300	1104.61	130.29	12.66
2-opt + Swapping + Insertion	100	1048.41	128.50	17.11

3.5.2 The Benefit of Using Multi-Compartment Vehicles

In this section, I illustrate the benefit of using multi-compartment vehicles instead of single compartment vehicles in garbage collection. In order to calculate this benefit, I solve the generated problem instances in two settings. In the first setting, two-compartment vehicles are used to collect the demand of all customers. In the second setting, only one compartment vehicles are used to collect the demand of all customers. The total vehicle capacity is kept same under both settings. Suppose the vehicle capacity of the compartments for two products are Q1 and Q2 under first setting. Then, the vehicle capacity under second setting is set to Q = Q1 + Q2

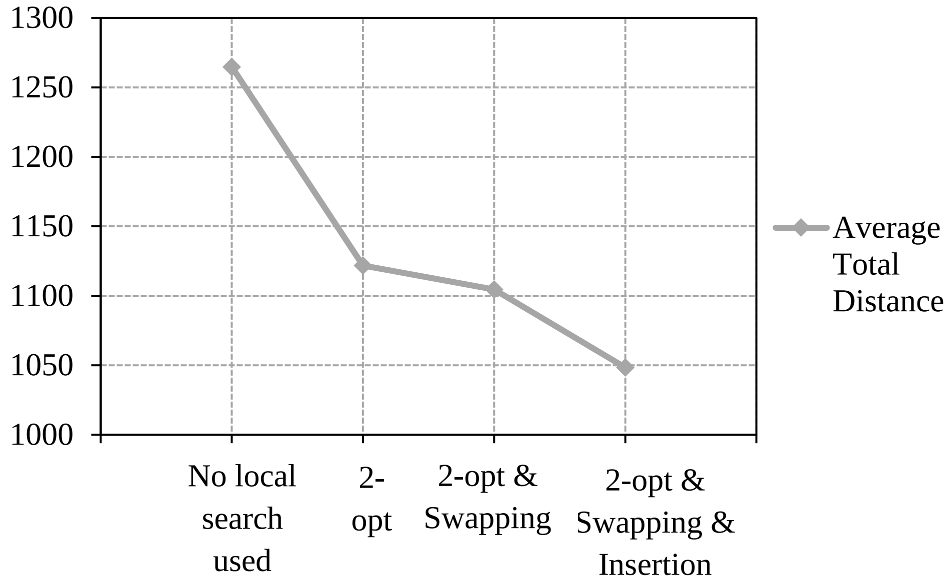

Figure 3.3: The effect of hybridization on the solution quality

In the second setting, only single compartment vehicles are used to collect the demand of all customers. Since the products have different characteristics, they cannot be transported within the same vehicle. Therefore, a customer is visited twice to collect product 1 and product 2 at different trips. In this case the problem is decomposed into two sub-problems. The first sub problem consists of solving the VRP for collection of product 1 and the second sub problem consists of solving VRP for collection of product 2. The sum of travelling cost for these two sub problems represents the solution for second setting. The results of 14 problem instances under two settings are presented in Table 3.3. The results reported under first setting (i.e., two compartments) are the same results reported in Table 3.1. The results reported under second setting (i.e., single compartment) are the combined tour length obtained after solving two sub problems using the proposed HAC.

Table 3.3: The benefit of using two-compartment vehicles instead of single-compartment vehicle

Problem	No. of customers	Two Compartment Total Length	Single Compartment Total Length	Percentage Total Length Increase
vrpnc1a	50	550.70	935.32	69.84
vrpnc1b	50	551.94	939.03	70.13
vrpnc2a	75	890.68	1319.33	48.13
vrpnc2b	75	918.96	1328.86	44.60
vrpnc3a	100	874.07	1432.56	63.90
vrpnc3b	100	895.26	1422.62	58.91
vrpnc4a	150	1126.12	1719.16	52.66
vrpnc4b	150	1159.48	1708.05	47.31
vrpnc5a	199	1444.29	2037.73	41.09
vrpnc5b	199	1525.87	2055.67	34.72
vrpnc6a	50	557.49	1113.36	99.71
vrpnc6b	50	559.37	1113.36	99.04
vrpnc7a	75	928.24	1802.44	94.18
vrpnc7b	75	932.67	1802.44	93.26
vrpnc8a	100	882.96	1749.80	98.17
vrpnc8b	100	884.85	1749.80	97.75
vrpnc9a	150	1228.88	2421.78	97.07
vrpnc9b	150	1226.58	2421.78	97.44
vrpnc10a	199	1511.65	2925.00	93.50

vrpnc10b	199	1526.02	2915.72	91.07
vrpnc11a	120	1110.45	1573.89	41.73
vrpnc11b	120	1221.73	1543.48	26.34
vrpnc12a	100	912.64	1279.78	40.23
vrpnc12b	100	950.79	1298.74	36.60
vrpnc13a	120	1556.46	3100.26	99.19
vrpnc13b	120	1550.12	3100.26	100.00
vrpnc14a	100	911.35	1643.49	80.34
vrpnc14b	100	965.84	1643.49	70.16
Average		1048.41	1789.19	70.97

It can be noticed that the total cost (total length) increases significantly when single compartment vehicles are used. In this case, a customer is visited twice to collect two different products. Visiting a customer twice increases the total travelling cost, although the travel cost per trip decreases due to the increase of vehicle capacity which allows vehicles to serve more customers per trip. The results reported in Table 3.3 show that the average total tour length increases by 70% when single compartment vehicles are used to serve the customers. The high percentage increase of tour length clearly exhibits the benefit of dividing the vehicles in different compartments instead of using the same vehicles with single compartment.

3.6 Conclusion

I propose a HAC algorithm combined with local search procedures to solve the MCVRP. I generate new bench mark problems to test the effectiveness of the proposed algorithm. The newly generated problem instances can also be used for testing other algorithms for the MCVRP in the future. The proposed HAC and the existing ACS are coded and numerical experiments are performed to evaluate the performance of the proposed HAC. The new algorithm improved the solutions in almost all instances. The numerical experiments indicate that on average, the proposed HAC algorithm produces better results using less computational time. In addition, it maintains its high performance in larger problems as well. The numerical results also indicated that hybridization is required to improve the performance of AC algorithm. Moreover, I illustrate the benefit of using multi-compartment vehicles instead of single-compartment vehicles.

Chapter 4

Travelling Salesman Problem for Online Orders in Retail Industry

4.1 Introduction

Routing is one of the main operational decisions in supply chain management. It deals with the transportation of goods between different supply chain components. This class of problems is concerned with designing the optimal routes for the delivery of goods between depots and customers. The VRP was formulated by Dantzig and Ramser [3] as a generalized form of the famous TSP. These two problems captured the attention of many researchers since the transportation process can denote 10% to 20% of the cost of any product [1]. The interest in optimizing the transportation process increased even more because it is one of the main consumers of fossil fuel which causes CO_2 emissions.

E-commerce is currently a vital tool for every successful business. It has been altering the retail industry and influencing markets behaviour. Companies are motivated to adopt different strategies and policies to cope up with this change in the market. One of the recently developed business models is called omni-channel retailing. Omni-channel retailing refers to the shopping experience in which there is no difference between online and regular shopping [93]. In fact, the retail industry has evolved from multi-channels toward omni-channels [5].

In omni-channel retailing, customers can get the products through different ways. They can order the products to be delivered at home or visit a store physically to get the products. Ordering the products may be done online or over the phone. For the products ordered online, the delivery of the products to the customers becomes the responsibility of the company. In order to increase the customer satisfaction, companies are working on making all the products available in stores as well as online. In addition, they are trying to reduce the time between orders placement and delivery times. This means that online orders should be satisfied from retail stores.

In this chapter, I investigate the TSP in omni-channel retail distribution systems. The problem is concerned with the design of the optimum route to deliver the products ordered online from the retail stores to the customers. The problem considers a retail company that owns a number of scattered retail stores in a specific area. The company offers a service for its customers by allowing them to order products online for home delivery. Each customer makes an order for a particular product provided that this product is available in at least one of the retail stores. The company is responsible for the delivery of products after deciding the retail store that will satisfy each customer order. Thus, the decisions of assigning customers to retail stores have been already made and know in advance. This means that each customer is assigned to one of the retail stores. Another application of the model may encounter a situation in which a third-party logistics company that provides delivery services for a number of small retail stores. The problem is NP-hard and is considered a generalization of the classical TSP. It reduces to classical TSP if all the retail stores are sharing the same location. In this case, that location is referred to as the depot. To the best of our knowledge, this problem has not been considered in the literature. However, problems that are closely related to the problem exist in the TSP literature. (i.e., the pickup and delivery travelling salesman problem (PDTSP).

The main contribution of this chapter is to introduce a new variant of the TSP that arises in omni-channel distribution systems. This problem has practical applications in most retail distribution systems. The mathematical formulation of the described problem is presented. In addition, I propose solution approaches to solve the problem. Newly generated bench mark problem instances are generated which can be used to compare the results of new solution approaches in future research work.

4.2 Literature Review

The TSP is the problem of determining the shortest Hamiltonian route. A Hamiltonian route starts from a given city, visits a number of cities once and only once, and returns back to the original city. It is one of the most studied yet challenging combinatorial optimization problems. Many exact algorithms have been proposed to solve the problem. Dantzig et al. [94] presented one of the earliest formulations to the problem. Little et al. [95] presented a branch and bound algorithm to solve the problem. Crowder and Padberg [96] presented a cutting plane approach coupled with branch and bound to solve the problem. The TSP is easy to describe and hard to solve. Thus, several heuristics have been tested by applying them to solve the problem. The SA was first tested to solve the TSP by Kirkpatrick et al. [79]. The GA was first applied to solve the TSP by Brady [71]. The memetic algorithm was proposed by Ulder et al. [97] to solve the TSP. The AC algorithm was first introduced to solve the TSP by Dorigo et al. [63]. Different surveys of the TSP can be found in [98-100].

The Pickup and Delivery Problem (PDP) is one of the most popular forms of the VRP due to its wide range of applications. It has been extensively studied in the literature of the TSP and the VRP as the PDTSP and PDVRP respectively. In the PDP, a number of transportation requests is satisfied by single or multiple vehicles. Each request is recognized by an origin and a destination. The route of the vehicle starts and ends at the depot. Lokin [43] introduced the precedence relations to the basic TSP. Kalantari et al. [44] presented a branch and bound algorithm for the single and multiple vehicle PDP. Savelsbergh & Sol. [45] presented a general model and survey for the PDP. Ruland and Rodin, [101] formulated the PDTSP as an integer program similar to the model of Dantzig et al. [94] and proposed a branch and cut algorithm to solve the problem. Renaud et al. [102] introduced a two-phase heuristic to solve the PDTSP. Renaud et al. [103] presented perturbation heuristics to solve the PDTSP. Dumitrescu et al. [104] proposed a branch and cut algorithm to solve the PDTSP. Hosny and Mumford [105] presented a GA, a SA, and a hill climbing algorithm to solve the PDTSP with time windows. Veenstra et al. [106] proposed a large neighborhood search heuristic to solve the PDTSP with handling cost. A number of survey papers was published recently dealing with the PDP [60-62].

4.3 Problem Description

The proposed model considers a set of c customers served by one vehicle which is available initially at the depot. Customers require deliveries of certain products from a set of r retail stores. Each customer requires demand for certain products to be supplied from a specific retail store. A basic assumption of the problem is that the retail store that satisfies the online order has been already decided. This means that the retail stores used for satisfying the demand of customers are known in advance. Every customer is associated with only one retail store that satisfies this customer demand.

A solution for the proposed model is a vehicle route that starts from the warehouse, visits a set of retail stores and customers in any sequence, and finally returns to the warehouse. However, the retail store must be visited before the associated customers. The proposed model aims to minimize the total cost of the vehicle routes such that

- The vehicle route starts and ends at the depot.
- Every retail store is visited only once.
- The total load carried by the vehicle remains under the capacity of the vehicle.
- Every customer is visited only once.
- Every customer is assigned to only one retail store.
- The customer is visited only after its associated retail store.

The chapter uses the following notations:

N Set of all nodes

R Set of retail stores nodes

C Set of customers nodes

d_j Online demand of customer *j*

p_j Quantity picked up at retail store *j*

C_{ij} The distance of traversing arc (*i*,*j*)

T_{ij} The time of traversing arc (*i*,*j*)

Q The vehicle capacity

O_i Service time at the location *i*

The problem can be defined using graph theory as follows:

Let $G = (V, A)$ be an undirected graph with a set of vertices $V = \{0, 1, \ldots, (r+c)\}$ where r is the number of the retail stores and c is the number of the customers. Node $\{0\}$ is the depot. Here, $R = \{1, \ldots, r\}$ are the retail stores, and $C = \{r+1, \ldots, r+c\}$ are the customers. Nodes $N = \{1, 2, \ldots, r, r+1, \ldots, r+c\}$ are all the nodes served by one vehicle (initially located in the depot). The total quantity carried by the vehicle does not exceed the capacity of the vehicle. Each node $j \in N$ has a quantity d_j to be delivered from a certain retail store (d_j equals zero for all retail stores). Each node $j \in N$ has a quantity p_j to be picked up at its location. No quantities are picked up at the customers locations (p_j equals zero for all customers). The term Y_{ij} is given and equal to 1 if customer j is assigned to retail store i and 0 otherwise. The term C_{ij} represents the travel distance (cost) for travelling from node i to node j. The term O_i represents the service time at location i. The term T_{ij} represents the time for travelling from node i to node j.

Let X_{ij} be binary flow variables equal to 1 if the arc (i j) is traversed by the vehicle, and 0 otherwise; Q_i be the load of the vehicle after serving node i; and S_i be the service start time of node i by the vehicle.

Mathematical Model

The proposed model can be formulated as follows:

Minimize

$$Z = \sum_{i \in V} \sum_{j \in V} C_{ij} X_{ij} \tag{4.1}$$

Subject to:

$$\sum_{j \in V} X_{ij} = 1 \qquad \forall i \in V, i \neq j \tag{4.2}$$

$$\sum_{i \in V} X_{ij} = 1 \qquad \forall j \in V, i \neq j \tag{4.3}$$

$$Q \geq Q_i \geq \left(Q_j + d_j - p_j\right) X_{ij} \qquad \forall i \in V, j \in N, i \neq j \tag{4.4}$$

$$S_j \geq (S_i + T_{ij} + O_i)\, X_{ij} \qquad \forall i \in V, j \in N, i \neq j \tag{4.5}$$

$$S_j \geq (S_i + T_{ij} + O_i)\, Y_{ij} \qquad \forall i \in R, j \in C \tag{4.6}$$

$$X_{ij} \in \{0,1\} \qquad \forall i \in V, j \in V, i \neq j \tag{4.7}$$

Eq. (4.1) is the objective function representing the total travelling cost of all arcs traversed by the vehicle. Eqs. (4.2-4.3) confirm that exactly one arc enters a node and exactly one arc leaves a node respectively. Eq. (4.4) confirms that the capacity of the vehicle is preserved. Eq. (4.5) confirms that the route connectivity is preserved and the sub tours are eliminated. Eq. (4.6) confirms that if customer *j* is assigned to retail store *i*, the retail store *i* is visited before the customer *j*.

The mathematical model is nonlinear. However, it is linearized to find optimal solutions for small size problems using CPLEX. To linearize the model, Eq. (4.4) is replaced by Eqs. (4.8-4.9) and Eq. (4.5) is replaced by Eq. (4.10).

$$Q \geq Q_i \qquad \forall i \in V \tag{4.8}$$

$$\left(Q_j + d_j - p_j - Q_i\right) \leq \left(1 - X_{ij}\right) M \qquad \forall i \in V, j \in N, i \neq j \tag{4.9}$$

$$(S_i + T_{ij} + O_i - S_j) \leq (1 - X_{ij})M \qquad \forall i \in V, j \in N, i \neq j \qquad (4.10)$$

Where M is a number with big value.

4.4 Solution Approaches

The problem is NP-hard which makes it difficult to solve the problem optimally. The mathematical model was solved using CPLEX solver in AMPL platform to get optimal solutions for small size problem instances. I propose two solution approaches (a heuristic and a metaheuristic) to solve larger problem instances. The next subsections provide more detailed description of these solution approaches.

4.4.1 Nearest-Neighbor Heuristic (NNH)

In this heuristic, the vehicle route is constructed using the minimum travel distance. The vehicle starts its route from the depot and visits the nearest retail store. After visiting a retail store the vehicle is loaded by the quantity required to satisfy the demand of the customers associated with this retail store. The vehicle then proceeds to the node with minimum distance (from the current location) among feasible nodes (i.e., retail store or customers). A retail store is considered feasible if there is enough capacity in the vehicle to pick up the products required by its associated customers. On the other hand, a customer is considered feasible if its associated retail store has been already visited by the vehicle. The vehicle proceeds from one node to another node till there are no feasible nodes available for visit. When there are no more feasible nodes to visit, the vehicle returns back to the depot.

4.4.2 Ant Colony Algorithm

The AC is an algorithm used to find near optimum solutions for larger problem instances of NP-hard problems. It is based on the behavior of the real ants when selecting the shortest route from their nest to the source of food. The algorithm is used in VRPs to construct routes by considering the vehicles as ants moving from one customer to another. Customers are chosen according to a probability function which is affected by two measures: the distance from current location to the next recommended location and the pheromone value. Pheromone is an indication of how frequently this arc was used to construct best solutions in previous iterations. The algorithm can be described in the following steps:

Step 1: Create initial solution to set initial pheromone values.

Step 2: Construct ant solutions for *m* number of ants.

Step 3: Improve the solutions produced by ants using local search.

Step 4: Update pheromone values on all arcs using best solution.

Step 5: Repeat the steps (2-4) until termination condition is reached and report the best solution.

4.4.2.1 Setting Initial Pheromone Values

In the beginning of the algorithm, initial pheromone values τ_{ij} are set on all arcs. Since there are no previous data exist, the pheromone values are set equal on all arcs. Initial solution is used to calculate the initial pheromone value which is set to the inverse of the total distance travelled to visit all nodes in the initial solution. I use the initial solution found in the NNH to calculate the initial pheromone values.

4.4.2.2 Solution Construction

In each iteration of the algorithm, *m* solutions are constructed. Each ant constructs a complete vehicle route. The ant starts its route from depot and visits a random retail store. The vehicle load is updated by adding the quantity required to satisfy the demand of the customers associated with this retail store. The ant selects the next node from a group of feasible nodes and adds it to the route. Feasible nodes are retail stores that there is enough capacity in the vehicle to pick up the products required by their associated customers. Customers that their associated retail stores have been already visited by the vehicle are considered feasible nodes as well. In order to select the next node, a random variable q is generated between $[0,1]$. If $q < q_{\circ}$, the feasible node with maximum attraction value $\varepsilon_{ij} = (\tau_{ij})^{\alpha}(\mu_{ij})^{\beta}$ is selected as the next node (where μ_{ij} is the inverse of the arc distance and τ_{ij} is the pheromone value on the arc). Otherwise, the next node is selected using the probability function in eq. (4.11).

$$P_{ij} = \begin{cases} \frac{\varepsilon_{ij}}{\sum_{l \in N_i} \varepsilon_{il}} & if\ j \in N_i \\ 0 & otherwise \end{cases} \tag{4.11}$$

where N_i is the list of feasible nodes and ε_{ij} is the attraction value.

Ants add nodes to the routes till no more nodes are left. The vehicle then returns to the depot and the solution is completed.

4.4.2.3 Relocation Local Search

After *m* routes are constructed, the solutions are improved using relocation local search. This local search improves the solution by searching for a better location for each node. Each node is removed from its original position and tested for insertion in every position without violating the feasibility of the route. The node is then relocated in the position that yields minimum total travel distance of the route. The feasibility of the route is preserved by ensuring that every customer is visited after visiting its associated retail store. In addition, the capacity constraint should not be violated. All nodes are tested for relocation. The process is terminated when no more improvements is possible through relocation.

4.4.2.4 Pheromone Update

In each iteration, after all ant routes are improved using the local search, the best solution found so far is reported. In addition, pheromone values are update on all arcs. The update process is performed in two steps: 1) the pheromone value is reduced on all arcs. 2) the pheromone value is increased on the arcs used in the best solution found so far only. Eq. (4.12) is used to calculate the new pheromone value for all arcs.

$$\tau_{ab}^{new} = \begin{cases} \rho \times \tau_{ab}^{old} + 1/L^{best} & if\ arc\ ab \in best\ route \\ \rho \times \tau_{ab}^{old} & otherwise \end{cases} \tag{4.12}$$

Where L^{best} is the total length of the best solution found so far.

4.5 Numerical Experiments

In this section, problem instances are generated and the performances of the proposed solution approaches are evaluated.

4.5.1 Parameters Setting

In this chapter, two solution approaches; the NNH and the AC algorithm, are used to solve the problem. The NNH produces single solution to the problem. The AC algorithm uses 1000 iterations to find the best solution. In each iteration, the AC algorithm constructs *m* number of solutions where *m* is equal to the number of retail stores. The AC algorithm uses other parameters with the following values: α=1, β=2, q_o=0.5, and ρ=0.9.

4.5.2 Data Generation

Three sets of problem instances are used to evaluate the performance of the solution approaches. The objective of the first set of problem instances is to evaluate the performance of the proposed solution approaches using the optimal solutions. Thus, the first set consists of 15 small size problem instances so that optimal solutions could be found by solving the mathematical model of the problem. The objective of the second set of problem instances is to compare the performance of the proposed solution approaches and test other approaches in the future. Thus, the second set consists of 15 larger size problem instances. The last set of problem instances is created by Dumitrescu et al. [104] and I use it to test the performance of the AC algorithm against the branch and cut algorithm in [104]. In the first set I use the number of retail stores equals to 3, 4, and 5; the number of customers equals to 6, 8, 10, 12, and 14. In the second set I use the number of retail stores equals to 6, 8, and 10; the number of customers equals to 30, 35, 40, 45, and 50. The capacity of the used vehicle is 25 and 50 units in the first and second set of problem instances respectively. In both sets of problem instances, the customers are assigned to retail stores randomly; the locations of the retail stores and customers are created randomly by generating *X* and *Y* coordinates between [0,50]. The demand of the customers is generated randomly between [1,5] unit.

4.5.3 Computational Results

The problem was solved using a heuristic, a metaheuristic, and CPLEX solver. The heuristic and metaheuristics were coded in C and solved using a server with four 2.1 GHz processors 16 core each with total of 256 GB of RAM. The mathematical model was coded using AMPL and CPLEX solver on Mac computer. The performance of the NNH and the AC algorithm is evaluated against the optimal solution found using CPLEX. The percentage deviation (PD) of the total travel distance is calculated using Eq. (4.13) where PD_i represents the PD in the total length of problem *i*. The term C_i is the total length found by CPLEX for problem *i* and the term S_i is the total length found by the AC algorithm or NNH for the same problem *i*.

$$PD_i = \frac{S_i - C_i}{C_i} \times 100\% \tag{4.13}$$

The results of the first set of problem instances are presented in Table 4.1. The CPU time taken by CPLEX to calculate the optimal solution is reported in seconds in the table. The average CPU time of CPLEX is 782.73 sec while the CPU time of the NNH and the AC algorithm is less than 1 sec. The average total distance calculated by CPLEX, the NNH, and the AC algorithm is 204.64, 204.93, and 237.16 unit respectively. It can be noticed that the AC algorithm reached the optimal solution in all problem instances. The AC algorithm succeeded to provide high performance while maintaining shorter CPU time. On the other hand, the average PD of the solution found by the NNH is 15%. The heuristic performed well in some problem instances and reached a minimum PD of 2.1%. However, it failed to perform well in other problem instances and reached a maximum PD of 40 %.

The performance of the NNH and the AC algorithm in the second set of problem instances is compared on the basis of the relative percentage deviation (RPD) of the total travel distance. The RPD is calculated using Eq. (4.14) where RPD_i represents the percentage deviation for problem *i*. The term A_i is the total length found by AC algorithm for problem *i* and the term N_i is the total length found by the NNH for the same problem *i*.

Table 4.1: The results of the AC and the NNH against the optimum solution

Problem	Number of Retail stores	Number of Customers	CPLEX		AC		NNH	
			Total Distance	CPU Time	Total Distance	PD %	Total Distance	PD%
1	3	6	163.63	1	163.63	0.00	188.33	15.09
2	3	8	173.37	1	173.37	0.00	243.26	40.31
3	3	10	156.48	1	156.48	0.00	162.64	3.94
4	3	12	231.69	7	231.69	0.00	250.03	7.92
5	3	14	239.76	1500	239.76	0.00	293.46	22.40
6	4	6	149.44	1	149.44	0.00	152.58	2.10
7	4	8	187.22	1	187.22	0.00	201.02	7.37
8	4	10	213.32	1	213.32	0.00	286.13	34.13
9	4	12	213.15	45	213.15	0.00	230.28	8.03
10	4	14	238.37	80	238.37	0.00	260.00	9.08

11	5	6	191.72	1	191.72	0.00	217.90	13.66
12	5	8	228.65	3	228.65	0.00	252.39	10.38
13	5	10	182.07	1	182.07	0.00	192.01	5.46
14	5	12	229.73	87	229.73	0.00	255.27	11.12
15	5	14	271.05	10011	271.05	0.00	372.11	37.28
Average			204.64	782.73	204.93	0.00	237.16	15.22

$$RPD_i = \frac{N_i - A_i}{A_i} \times 100\% \qquad (4.14)$$

Table 4.2: The results of the NNH against the AC

Problem	Number of Retail stores	Number of Customers	AC		NNH	
			Total Distance	CPU Time	Total Distance	RPD%
1	6	30	357.70	13	475.06	32.81
2	6	35	348.50	22	505.95	45.18
3	6	40	382.73	36	570.96	49.18
4	6	45	427.46	49	498.62	16.65
5	6	50	436.09	77	521.49	19.58

6	8	30	293.94	30	386.93	31.64
7	8	35	354.75	37	455.77	28.47
8	8	40	385.82	63	471.97	22.33
9	8	45	430.70	85	542.33	25.92
10	8	50	449.16	125	684.69	52.44
11	10	30	348.52	35	437.75	25.60
12	10	35	374.07	60	504.22	34.79
13	10	40	361.17	77	527.88	46.16
14	10	45	436.48	114	628.10	43.90
15	10	50	516.46	135	656.07	27.03
Average			393.57	63.87	524.52	33.45

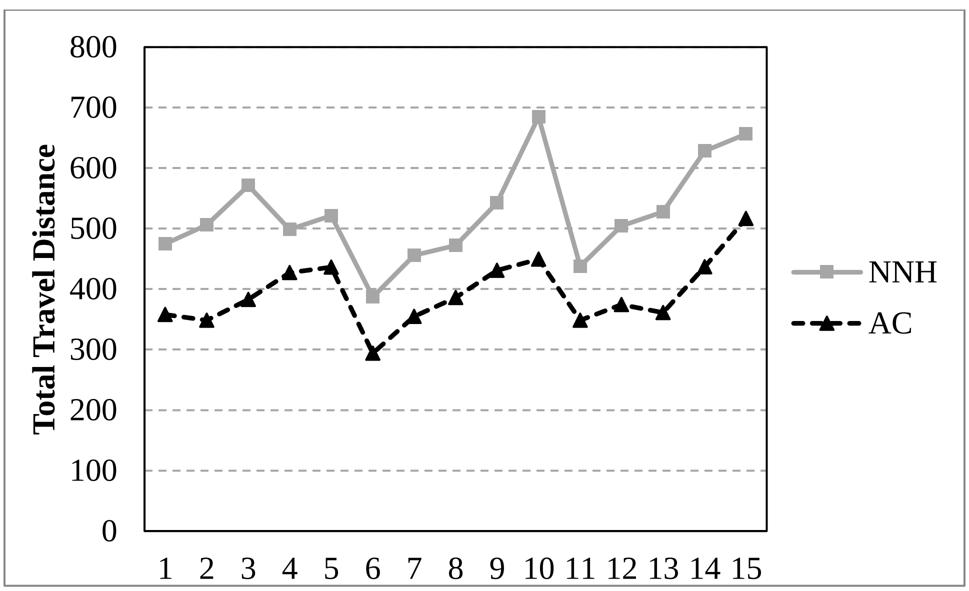

Figure 4.1. The results of the second set of problem instances using the AC and the NNH

The results of the second set of problem instances are presented in Table 4.2 and Figure 4.1. The CPU time taken by the AC algorithm is reported in seconds in the table. The average CPU time of the AC algorithm is 63.87 sec while the CPU time of the NNH is less than 1 sec. The average total distance calculated by the NNH and the AC algorithm is 524.52 and 393.57 unit respectively. It can be noticed that the AC algorithm performed better than the NNH in all problem instances. The average RPD of the NNH with respect to the AC algorithm is 33.45%. This was expected from the results of the first set of problem instances. However, it can be concluded that the AC algorithm maintained high performance in larger size problem instances.

The results of the third set of problem instances are presented in Table 4.3. It can be noticed that the AC algorithm reached the optimal solution in 24 problem instances and the best solution found by the branch and cut algorithm of Dumitrescu et al. [104] in another problem instance. The average deviation of the AC algorithm with respect to the branch and cut algorithm is 0.47%. The average CPU time of the AC algorithm is 36 seconds while the average CPU time of the branch and cut algorithm is 3606 seconds. It can be concluded that the AC algorithm succeeded to provide high performance while maintaining shorter CPU time.

Table 4.3: The results of the AC against the branch and cut

Problem	Number of requests	Branch & Cut		AC		
		Total Distance	CPU Time	Total Distance	CPU Time	Deviation %

1	5	3585	0	3585	0	0
2	5	2565	0	2565	0	0
3	5	3787	0	3787	0	0
4	5	3128	0	3128	0	0
5	5	3123	0	3123	0	0
6	10	4896	3	4896	0	0
7	10	4490	2	4490	0	0
8	10	4070	0	4070	0	0
9	10	4551	1	4551	0	0
10	10	4874	4	4874	0	0
11	15	5150	8	5150	2	0
12	15	5391	21	5391	2	0
13	15	5008	0	5008	2	0
14	15	5566	14	5566	2	0
15	15	5229	0	5229	2	0
16	20	5698	12	5698	10	0
17	20	6213	20	6213	9	0
18	20	6200	19	6200	8	0
19	20	6106	17	6106	9	0
20	20	6465	58	6465	9	0
21	25	7332	14400	7345	23	0.18
22	25	6665	3138	6807	28	2.13

23	25	7095	291	7095	30	0
24	25	7069	14323	7075	27	0.08
25	25	6754	72	6754	28	0
26	30	7309	14400	7456	78	2.01
27	30	6857	2843	6857	64	0
28	30	7723	1891	7723	61	0
29	30	7310	573	7390	69	1.09
30	30	7213	14400	7346	69	1.84
31	35	7746	2104	7971	146	2.90
32	35	7904	14400	8098	161	2.45
33	35	7949	14400	8205	166	3.22
34	35	7905	14400	7905	133	0
35	35	8530	14400	8564	144	0.40
Average		5927.31	3606.11	5962.46	36.63	0.47

4.6 Conclusion

In this Chapter, I introduce the TSP arising in omni-channel retail distribution systems. A mathematical model is presented to describe the problem. The mathematical problem is solved using CPLEX solver to find optimum solutions. A nearest neighbor heuristic and an AC algorithm are proposed to solve the problem. Bench mark problem instances are generated to test the performance of the proposed algorithms. The AC algorithm showed good performance with respect to the optimal solutions found using the CPLEX. It reached the optimal solution of all problem instances. It also succeeded to provide high performance while maintaining shorter CPU time against the branch and cut algorithm.

Chapter 5

Multi-Compartment Vehicle Routing Problem for Omni-Channel Retail Distribution

5.1 Introduction

"There was a time when the online and offline businesses were viewed as being different," said Walmart.com Chief Executive Raul Vazquez. "Now we are realizing that we actually have a physical advantage thanks to our thousands of stores, and we can use it to become No. 1 online." [107]. The statement of Walmart executive highlights the importance of online business in retail industry. In fact, Walmart delivered 40 percent of its online orders through their stores in that year. According to the National Retail Federation report [108], the business of 3.8 million retail establishments performed 12 percent of all business establishments in US in year 2012. The retail industry provided 42 million jobs which was 23.4 percent of the total national employment in US. The total labor income generated by the retail industry was $1.58 trillion which was 16.1 percent of the total labor income. The total GDP impact of the retail industry was $2.59 trillion which was 16 percent of the total GDP.

According to the Global Powers of Retailing report [4], the 250 largest retailers around the world generated revenues of US$4.31 trillion in 2015. This figure indicates that despite the challenging global economy in 2015, retailers managed to keep steady growth and international expansion. The revenue generated by e-commerce was 8.7 percent of the total retail revenue of the top 250 retailing companies. The increasing share of e-commerce in retail industry clearly exhibits the shift of retail industry from single channel to omni-channels [5]. The omni-channel system is the most recent business model used in the retail industry. The new business model aims to increase the sales volume as well as the customer satisfaction in retail industry. It provides the customers with the flexibility to get the products through different channels. Customers can choose to get the products by physically visiting the store, or by ordering the products online for home delivery. The research conducted by Herhausen et al. [109] showed that the online-offline channel integration not only enhances the shopping experience of the customers but it also provides competitive advantages for the organization.

This chapter considers the VRP arising in omni-channel retailing. The considered problem studies the distribution network of a giant retailing company operating in different cities. In a given city, it owns a central warehouse and a group of retail stores. The proposed model considers the distribution system for this giant retailing company in a particular city only. The company offers its consumers the choice of purchasing products in two ways; visiting the stores physically or ordering the products online. The company offers a group of products that can be ordered online. A consumer places an order for a specific product available in one of the retail stores and the products are supplied to the consumer from the chosen retail store. Thus, I assume that a decision on the retail stores that satisfies the demand of consumers has been already made. In other words, the retail stores used for satisfying the demand of consumers are known in advance. This assumption is supported by the statement of Walmart executive which states that 40 percent of their online orders are delivered through their stores.

The distribution network of the company performs two types of deliveries; deliveries from the warehouse to the retail stores and deliveries from the retail stores to the consumers. The company tries to utilize the same fleet of vehicles to serve the retail stores and the consumers. I consider that the products required by the consumers will be stored in different compartment of the same vehicle. Thus, handling the products required by the consumers does not interfere with the products required by the retail stores. The fleet of vehicles is available at the warehouse.

Our model suggests that both retail stores and consumers are visited using the same fleet of vehicles. Hence, the proposed model is applicable for the organizations that use light to medium trucks in serving their scattered retail stores. Several cities have imposed restrictions on the times and days when heavy trucks are allowed inside the city. These restrictions are applied on vehicles exceeding the gross weight of 7.5 tons which produce more noise and pollution. For example, according to [110] trucks are allowed to use selected routes only in Winnipeg, Canada. In this case, organizations prefer to use smaller trucks to have the privilege of using their fleet without any restrictions. The proposed model is applicable in this situation as well. The proposed model is not applicable in some business models where only large trucks are used to deliver products to the retail stores.

In this chapter, I introduce of a new variant of the VRP arising in the omni-channel retail distribution networks. In addition, the chapter presents a mathematical formulation for the problem to obtain the optimal solutions. Moreover, the chapter proposes different heuristics and a metaheuristic to solve large scale problems. Finally, new bench mark problem instances are introduced for testing the proposed algorithms and for testing algorithms in future work. The presented model in this chapter is applicable in retail stores such as: Shoppers Drug Mart, Walmart, etc.

5.2 Literature Review

The considered problem in this chapter is closely related to the CVRP and the PDP. Thus, the literature related to this problem is divided into two directions: literature related to the CVRP and literature related to the PDP. The CVRP was considered as a generalization of the TSP by Dantzig & Ramser [3]. This was the first introduction of the VRP in the operations research literature. Afterwards several variants of the VRP evolved in the operations research literature. Different solution approaches were proposed to solve the problem. A review of the most recent exact algorithms focused on the CVRP can be found in Baldacci et al. [111]. Recently, new formulations for the problem and new lower bounds were presented by Letchford and Salazar-Gonzalez [10]. The exact algorithms can solve only small problem instances. Therefore, heuristic methods are considered a popular choice for solving the real life VRPs. Different heuristics have been used in the VRP literature to find near optimum solutions (such as: AC algorithm (Mazzeo and Loiseau

[112]), hill climber heuristic (Derigs & Kaiser [11]), memetic algorithm (Nagata & Bräysy [13]), PSO (Ai & Kachitvichyanukul [12]), and artificial bee colony (Szeto et al. [14])). For recent research in VRP and its variants see [21-23].

The PDP is one of the most popular variant of the VRP due to its wide range of applications. It has been extensively studied in the literature. The precedence relations of the PDP were introduced to the basic TSP by Lokin [43]. A branch and bound algorithm for the single and multiple vehicles PDP was presented by Kalantari et al. [44]. The first general model and survey for the problem were introduced by Savelsbergh & Sol. [45]. A mixed integer linear programming formulation for the multiple vehicles PDP was presented by Lu, & Dessouky [46]. They developed a branch and cut algorithm to solve the problem. A number of survey papers is published recently dealing with the PDP [60-62]. The proposed problem in this chapter combines the structure of the CVRP and PDP along with the two compartment feature. To the best of our knowledge, the considered problem in this chapter has not been tackled in the literature.

The proposed problem falls under the category of NP-hard problems which is difficult to solve using exact algorithms especially with large problem instances. Therefore, it is common in the VRP literature to consider heuristics and metaheuristics to solve different variants of the VRP. In this chapter, I present different heuristics to solve the proposed problem. In addition, I use an AC algorithm to solve the problem. The AC algorithm was first introduced to solve combinatorial problems by Dorigo et al. [63]. It was applied to solve the TSP. The first application of the AC for solving the VRP was proposed by Bullnheimer et al. [64]. Although the initial experiments in [64] did not improve the best known solutions, it provided promising results. The AC produced good results when it was used to solve the dynamic VRP by Montemanni et al. [67]. An improved AC was introduced to solve the VRP with backhauls in [36] and to solve the VRP with simultaneous delivery and pickup in [37]. The AC was used to solve the multi-compartment VRP in [66]. A multiple AC system was proposed for the VRP with time windows and uncertain travel times in [17].

5.3 Problem Description and Formulation

In this section, I provide a brief description of the problem and present its mathematical formulation. The considered problem includes two sets of customers served by a fleet of homogenous vehicles. The first set of customers consists of r retail stores. The second set of customers consists of c consumers. The first set includes the customers who require some products to be delivered from a distribution center. The second set includes the consumers who require products to be supplied from specific retail store. The fleet of vehicles is available at the depot. Every consumer is associated with only one retail store that satisfies its demand. Every vehicle has two compartments reserved for the retail stores demand and consumers demand. In this chapter, the two compartments are referred as retail store compartment and consumer compartment respectively.

A solution for the proposed problem consists of number of vehicle routes. Each vehicle starts from the warehouse, visits a set of retail stores and consumers in any sequence, and finally returns back to the warehouse. However, the retail store must be visited before the associated consumer. The proposed model aims to minimize the total cost of the vehicle routes such that:

- Every route starts and ends at the depot.
- Every retail store is visited by only one vehicle and only once.
- The total load arising from the demand of the retail stores remains under the capacity of the retail store compartment.
- The total load arising from the demand of the consumers remains under the capacity of the consumer compartment.
- Every consumer is visited by only one vehicle and only once.
- Every consumer is assigned to only one retail store.
- The consumer and its associated retail store are visited by the same vehicle.
- The consumer is visited only after visiting its associated retail store.

Notations:

N Set of all customer nodes

R Set of retail stores

C Set of online consumers (consumers ordering products online)

K Set of vehicles

M Set of compartments

D_j Demand delivered from the depot to customer *j*

d_j Online demand of customer *j*

p_j Quantity picked up at customer *j*

C_{ij} The distance of traversing arc (*i*,*j*)

T_{ij} The time of traversing arc (*i,j*)

Q_1 The vehicle capacity of retail store compartment

Q_2 The vehicle capacity of consumer compartment

O_i Drop time at the location *i*

The problem can be defined using graph theory as follows:

Let $G = (V, A)$ be an undirected graph with a set of vertices $V= \{0, 1,\ldots, (r+c)\}$ where r is the number of the retail stores and c is the number of the consumers. Node {0} is the distribution center from where products are delivered to retail stores. Here, $R= \{1, \ldots, r\}$ are the retail stores, and $C= \{r+1,\ldots, r+c\}$ are the consumers. Nodes $N= \{1, 2, \ldots, r, r+1, \ldots, r+c\}$ are the customer nodes (retail stores and consumers) served by a number of K identical vehicles (initially located at the depot). Each vehicle has 2 compartments (one for retail stores demand and the other for the online demand of the consumers). The total quantity carried in each of the compartments does not exceed the capacity of the compartments (Q_1, and Q_2). Each customer $j \in V$ has a quantity D_j to be delivered from the depot (D_j equals zero for all consumers). Each customer $j \in V$ has a quantity d_j to be delivered from a certain retail store (d_j equals zero for all retail stores). Each customer $j \in V$ has a quantity p_j to be picked up at his location. No quantities are picked up at the consumers locations (p_j equals zero for all consumers). Y_{ij} is equal to 1 if consumer j is assigned to retail store i and 0 otherwise. C_{ij} represents the travel distance (cost) for travelling from node i to node j. O_i represents the drop time at location i. The term T_{ij} represents the time for travelling from node i to node j.

Let X_{ij}^{k} be binary flow variables equal to 1 if the arc (i j) is traversed by vehicle k, and 0 otherwise; $Q1_{i}^{k}$ be the load of compartment 1 (retail store compartment) after serving node i; $Q2_{i}^{k}$ be the load of compartment 2 (consumer compartment) after serving node i; and S_{i}^{k} be the service start time of node i for vehicle k. The proposed model can be formulated as follows:

Minimize

$$Z = \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} C_{ij}^{k} X_{ij}^{k} \qquad (5.1)$$

Subject to:

$$\sum_{k \in K} \sum_{j \in V} X_{ij}^{k} = 1 \qquad \forall i \in N, i \neq j \qquad (5.2)$$

$$\sum_{j \in R} X_{(0)j}^{k} = 1 \qquad \forall k \in K \qquad (5.3)$$

$$\sum_{i \in V} X_{ij}^{k} - \sum_{i \in V} X_{ji}^{k} = 0 \qquad \forall j \in N, k \in K, i \neq j \qquad (5.4)$$

$$\sum_{i \in N} X_{i(0)}^{k} = 1 \qquad \forall k \in K \qquad (5.5)$$

$$Y_{ij} \left(\sum_{l \in N} X_{il}^{k} - \sum_{l \in N} X_{lj}^{k}\right) = 0 \qquad \forall i \in R, j \in C, k \in K, i \neq l, j \neq l \qquad (5.6)$$

$$Q_{1} \geq Q1_{i}^{k} \geq \left(Q1_{j}^{k} + D_{j}\right) X_{ij}^{k} \qquad \forall i \in V, j \in N, k \in K, i \neq j \qquad (5.7)$$

$$Q_{2} \geq Q2_{i}^{k} \geq \left(Q2_{j}^{k} + d_{j} - p_{j}\right) X_{ij}^{k} \qquad \forall i \in V, j \in N, k \in K, i \neq j \qquad (5.8)$$

$$S_{j}^{k} \geq (S_{i}^{k} + T_{ij} + O_{i})\, X_{ij}^{k} \qquad \forall i \in V, j \in N, k \in K, i \neq j \qquad (5.9)$$

$$S_{j}^{k} \geq (S_{i}^{k} + T_{ij} + O_{i})\, Y_{ij} \qquad \forall i \in R, j \in C, k \in K, i \neq j \qquad (5.10)$$

$$X_{ij}^{k} \in \{0,1\} \qquad \forall i \in V, j \in V, i \neq j, k \in K \qquad (5.11)$$

Eq. (5.1) is the objective function representing the total travelling cost of all arcs traversed by all vehicles. Eq. (5.2) confirms that exactly one arc enters a customer node.

Eqs. (5.3-5.5) confirm that each vehicle starts its route from the depot and ends its route at the depot. Eq. (5.6) confirms that if a consumer *j* is assigned to a retail store *i*, both the retail store and consumer are visited by the same vehicle. Eqs. (5.7-5.8) confirm that the total load of the vehicle is under the capacity of the compartments. Eq. (5.9) confirms that the time consistency is conserved. Eqs. (5.7-5.9) confirm that the sub tours are eliminated. Eq. (5.10) confirms that if a consumer *j* is assigned to a retail store *i*, the retail store *i* is visited before the consumer *j*.

The mathematical model is nonlinear and the problem falls under the category of NP-hard problems. However, the mathematical model was linearized and solved using CPLEX solver in AMPL platform to get optimal solutions for small size problem instances. The optimal solutions found by CPLEX are used to evaluate the performance of the proposed solution approaches. The model is linearized by replacing Eq. (5.7) with Eqs. (5.12 and 5.13), replacing Eq. (5.8) with Eqs. (5.14-5.15), and replacing Eq. (5.9) with Eq. (5.16) respectively.

$$Q_1 \geq Q1_i^k \qquad \forall i \in V, k \in K \tag{5.12}$$

$$\left(Q1_j^k + D_j - Q1_i^k\right) \leq \left(1 - X_{ij}^k\right)M \qquad \forall i \in V, j \in N,, k \in K \;\; i \neq j \tag{5.13}$$

$$Q_2 \geq Q2_i^k \qquad \forall i \in V, k \in K \tag{5.14}$$

$$\left(Q2_j^k + d_j - p_j - Q2_i^k\right) \leq \left(1 - X_{ij}^k\right)M \qquad \forall i \in V, j \in N,, k \in K \;\; i \neq j \tag{5.15}$$

$$(S_i^k + T_{ij} + O_i - S_j^k) \leq \left(1 - X_{ij}^k\right)M \qquad \forall i \in V, j \in N, k \in K, i \neq j \tag{5.16}$$

Where *M* is a number with big value.

5.4 Solution Approaches

In addition to the mathematical model, this chapter presents three solution approaches (two heuristics and one metaheuristic) to solve the problem. These solution approaches are described in the next subsections.

5.4.1 Insertion Heuristic (IH)

The insertion heuristic finds the solution in two phases. In the first phase, a vehicle route consisting of retail stores only is constructed on the basis of the nearest neighbor criteria. The retail store compartment capacity is preserved while building the routes. In the second phase, online consumers are added to the constructed routes. Consumers can be added in any position of the route after the position of their associated retail store. The consumer compartment should be preserved while inserting the consumers. Consumers are inserted one after another in the feasible position that minimizes the total distance of the route. The process continues till all consumers are assigned to vehicle routes.

5.4.2 Minimum Weighted Distance Heuristic (MWD)

In this heuristic, I construct solutions in one phase by adding retail stores and consumers to vehicle routes simultaneously. Each vehicle starts from depot and visits feasible nodes (retail stores and consumers) on the basis of the minimum weighted distance criteria. A retail store node is considered feasible if it satisfies two constraints: 1) its retail store demand does not violate the capacity constraint of compartment 1; 2) the demand of its associated consumers does not violate the capacity constraint of compartment 2. A consumer node is considered feasible if its associated retail store has been already visited by the current vehicle. The vehicle proceeds from one customer to another till there is no feasible node available for visiting. The vehicle then returns back to the depot and another trip starts from the depot to serve the remaining nodes.

The distinct feature of this heuristic is the consideration of weighted distance between nodes. The weighted travel distance is obtained by multiplying the actual distance to the next feasible consumer by a factor x. The weighted distance for retail stores nodes is kept as the actual distance. The weighted distance is used to select which node is visited from the current location of the vehicle. The value of factor x determines the preference between consumers and retail stores. If the value of factor x is less than one, this means that consumers have the preference to be visited over retail stores. If the value of factor x is more than one, this means that retail stores have the preference to be visited over consumers. I construct different solutions for each problem by multiplying the distances to the online consumers by different values of the factor x (0.5, 0.75, 1, 1.25, and 1.5). Finally, I choose the solution that has the minimum total travel distance among different values of factor x.

5.4.3 Ant-colony Algorithm

The AC is an algorithm used in solving NP-hard problems by finding near optimum solutions. It was inspired from the behavior of the real ants in finding the best route from their nest to the source of food. The algorithm is widely used in VRPs to construct routes due to its efficiency in finding good solutions in reasonable time. Customers are added to the current vehicle route according to a probability function composed of two parts: the distance from current customer to the next customer and the pheromone value. Pheromone is the desirability of an arc measured by how often this arc is chosen in previous solutions. The algorithm can be described in the following steps:

Step 1: Create initial solution to initialize the trail intensities.

Step 2: Construct *m* complete tours using *m* number of ants.

Step 3: Improve the solution produced by each ant using Intra-route local search.

Step 4: Update pheromone values on all arcs using best solution.

Step 5: If the termination condition is not met: go back to step 2

Step 6: Report the best solution.

5.4.3.1 Deposit Initial Pheromone Values

The pheromone value τ_{ij} indicates the frequency of selecting arc (*ij*) in previous solutions. In the beginning of the algorithm, the initial pheromone values are set equally for all arcs. The initial pheromone value is equal to the inverse of the total travelled distance calculated from an initial solution. I use the MWD heuristic to initialize pheromone.

5.4.3.2 Route Construction

I use *m* ants to construct complete tours. Each ant starts from the depot and chooses a feasible node randomly. Then, the next node is chosen from the list of feasible nodes as described in section 5.4.2. In order to choose the next node, a random variable q is calculated between $[0,1]$. If $q < q_\circ$, the next chosen node is the feasible customer with maximum attraction value $\varepsilon_{ij} = \left(\tau_{ij}\right)^{\alpha}\left(\mu_{ij}\right)^{\beta}$ where μ_{ij} is the inverse of the arc distance. Otherwise, the next node is chosen according to the probability function in eq. (5.17).

$$P_{ij} = \begin{cases} \dfrac{\varepsilon_{ij}}{\sum_{l \in N_i} \varepsilon_{il}} & if\ j \in N_i \\ 0 & otherwise \end{cases} \tag{5.17}$$

where N_i is the list of feasible nodes and ε_{ij} is the attraction value.

Customers are added to the vehicle route till no more nodes can be added feasibly to the vehicle route. The vehicle returns back to the depot and a new trip is started. The route is completed when all customers (retail stores and consumers) are visited.

5.4.3.3 Intra-Trip Local Search

In the route construction phase *m* solutions are found using *m* ants. In this phase, the *m* solutions are improved using the local search. This local search improves the solution of each trip by repositioning the nodes. Nodes are removed from their original position and tested for insertion in every feasible position in the same trip. However, the position that minimizes the total travel distance is chosen. The repositioning of a node in a certain position is considered feasible if it does not violate the sequence constraint or the capacity constraint. The sequence constraint is violated if a consumer is located before its associated retail store.

5.4.3.4 Pheromone Update

After local search is performed, the distance is updated for all routes. The best solution found so far is used to update the pheromone values using eq. (5.18). Updating trail intensities is described in two steps: lowering of pheromone on all arcs and increasing on the arcs reported in the best route.

$$\tau_{ij}^{new} = \begin{cases} \rho \times \tau_{ij}^{old} + 1/L^{best} & if\ arc\ ij \in best\ route \\ \rho \times \tau_{ij}^{old} & otherwise \end{cases} \tag{5.18}$$

where L^{best} is the total length of the best solution in each iteration.

5.5 Numerical Experiments

In next subsections, I present the data generation method to perform the numerical experiments. The numerical experiments are used to compare the performance of the different proposed solution approaches.

5.5.1 Parameter Setting for the AC Algorithm

The above described AC algorithm uses number of parameters. I set these parameters on the basis of the recommended values in the literature. The solution quality of the AC depends mainly on the number of iterations and the number of used ants. Increasing the number of iterations and the number of ants increases the solution quality. However, it also increases the computational time. I used 1000 iterations and a number of ants equals to the number of retail stores to keep the CPU time reasonable. I use (α=1 and β=2) and set (q_o=0.9). I use (ρ=0.9) to update the pheromone with new and existing experiences of ants.

5.5.2 Data Generation

There are no bench mark problems for VRP in omni-channels because the problem is introduced for the first time in this chapter. In this work, I use two sets of problem instances: small size problem instances and large size problem instances. The small problem instances are used to compare the solution of the proposed solution approaches with the optimal solution obtained by solving the mathematical model. This comparison gives an indication about the performance of the proposed heuristics and metaheuristic. The large problem instances are used to compare the performance of the proposed heuristics and metaheuristic approaches. These later cases can be used for testing the performance of any proposed solution approaches in the future. The X and Y coordinates of retail stores and consumers are created randomly between [0,100]. The demand of the retail stores is generated between 25 and 50 units. The retail store compartment capacity is set to 100 units and the consumer compartment capacity is set to the maximum pickup quantity from any retail store. The list of served consumers for each retail store is generated randomly. In the first set, I use number of retail stores of 3, 4, 5, and 6, number of consumers of 6, 9, 12, and 15. Thus, a total of 16 problem instances are generated. In the second set, I use number of retail stores of 10, 15, 20, and 25, number of consumers of 25, 50, 75, 100, and 150. Thus, a total of 20 problem instances are generated.

5.5.3 Computational Results

The proposed heuristics and metaheuristic were coded in C programming language and the 36 problem instances were solved. The problems were solved using a server that operates four 2.1GHz processors with 16-core each and a total of 256 GB of RAM. The mathematical model was coded using AMPL and solved using CPLEX solver. The performances of the three solution approaches are evaluated with respect to their solutions using the PD. The PD is calculated according to Eq. (5.19), where PD_i represents the percentage deviation in the total length of problem i. Furthermore, the term C_i is the total length found by CPLEX for problem i and the term S_i is the total length found by each heuristic and metaheuristic for the same problem i.

$$PD_i = \frac{S_i - C_i}{C_i} \times 100\% \tag{5.19}$$

The results of small problem instances are presented in Table 5.1. The CPU time taken by the AC algorithm and the two heuristics was less than one second. It can be noticed that the AC algorithm was able to reach the optimum solution found by the CPLEX in 14 out of the 16 problem instances. The average PD of the AC with respect to the optimum solution found by CPLEX is 0.10%. It can be concluded that the AC algorithm has a good performance while it succeeded to keep shorter CPU time compared to the solution found by CPLEX which took 1217 seconds on average. On the contrary, the performances of the two heuristics are poor with an average of 14.7% and 12.50% PD with respect to the solution found by the CPLEX for the MWD and IH respectively. The performances of the two heuristics are so close but on average the IH has a slightly better performance.

Table 5.1: The results of the AC, the MWD, and the IH against the optimum solution

Problem	Number of Retail stores	Number of Consumers	CPLEX		AC		MWD		IH	
			Total Distance	CPU Time	Total Distance	PD%	Total Distance	PD%	Total Distance	PD%
1	3	6	473.38	1	473.38	0.00	473.62	0.05	473.62	0.05
2	3	9	545.64	1	545.64	0.00	596.20	9.27	589.54	8.05
3	3	12	521.95	1	521.95	0.00	557.05	6.73	560.70	7.42
4	3	15	557.42	3	557.42	0.00	639.30	14.69	707.86	26.99
5	4	6	475.08	1	475.08	0.00	608.83	28.15	650.10	36.84
6	4	9	514.68	4	514.68	0.00	553.23	7.49	553.23	7.49
7	4	12	517.14	7	517.14	0.00	573.84	10.96	554.68	7.26
8	4	15	518.21	2	518.21	0.00	577.88	11.51	597.00	15.20
9	5	6	526.20	12	526.20	0.00	573.83	9.05	526.20	0.00
10	5	9	650.71	55	650.71	0.00	854.72	31.35	788.12	21.12
11	5	12	696.22	1786	700.57	0.62	877.97	26.11	737.19	5.88

12	5	15	771.49	1968	779.59	1.05	1012.45	31.23	906.72	17.53
13	6	6	584.87	4	584.87	0.00	650.63	11.24	650.63	11.24
14	6	9	603.22	13	603.22	0.00	674.47	11.81	674.19	11.77
15	6	12	747.80	1118	747.80	0.00	793.80	6.15	793.80	6.15
16	6	15	987.07	14504	987.07	0.00	1180.56	19.60	1155.06	17.02
Average			605.69	1217	606.47	0.10	699.90	14.71	682.41	12.50

For large problem instances, the performances of the three solution approaches are evaluated with respect to the best solution found using the relative percentage deviation (RPD). The RPD is calculated according to Eq. (5.20), where RPD_i represents the relative percentage deviation in the total length of problem *i*. Furthermore, the term B_i is the best total length found by any of the solution approaches for problem *i* (the AC algorithm gave the best solution in all problems). The term S_i is the total length found by the other heuristics for the same problem *i*.

$$PD_i = \frac{S_i - B_i}{B_i} \times 100\% \tag{5.20}$$

The results of large problem instances are presented in Table 5.2. The average computational time for the heuristics is 0.1 seconds while the average computational time for the AC algorithm is 22.6 seconds. The total lengths calculated using the AC algorithm, the MWD and the IH respectively are presented in the table and shown in Figure 5.1. It is obvious that the results obtained from the AC algorithm are better than the results obtained using other heuristics in all problem instances.

Table 5.2: The results of the MWD and the IH against the AC algorithm

Problem	Number of retail stores	Number of consumers	Total Distance AC	MWD		IH	
				Total Distance	RPD%	Total Distance	RPD%
1	10	25	1253.12	1383.97	10.44	1473.43	17.58
2	10	50	1509.58	1683.89	11.55	1723.83	14.19
3	10	75	2231.93	2694.14	20.71	2790.28	25.02
4	10	100	2272.95	2753.82	21.16	2753.66	21.15
5	10	150	2625.62	3333.66	26.97	3402.43	29.59
6	15	25	1533.00	1789.53	16.73	1749.29	14.11
7	15	50	2022.89	2401.73	18.73	2456.19	21.42
8	15	75	2512.35	3274.76	30.35	2841.17	13.09
9	15	100	3079.58	3479.57	12.99	3698.46	20.10
10	15	150	3718.84	4431.01	19.15	4380.79	17.80
11	20	25	1900.56	2326.96	22.44	2226.39	17.14
12	20	50	2333.18	2935.94	25.83	2992.87	28.27
13	20	75	2902.04	3424.07	17.99	3573.05	23.12
14	20	100	3573.64	4222.43	18.15	4482.18	25.42
15	20	150	4391.81	5220.95	18.88	4875.63	11.02
16	25	25	2086.49	2338.75	12.09	2392.80	14.68
17	25	50	2610.70	3095.03	18.55	3100.54	18.76
18	25	75	3186.29	3724.44	16.89	3597.55	12.91
19	25	100	3920.02	4513.71	15.15	4579.02	16.81

20	25	150	4491.01	5370.92	19.59	5574.92	24.14
Average			2707.78	3219.96	18.92	3233.22	19.40

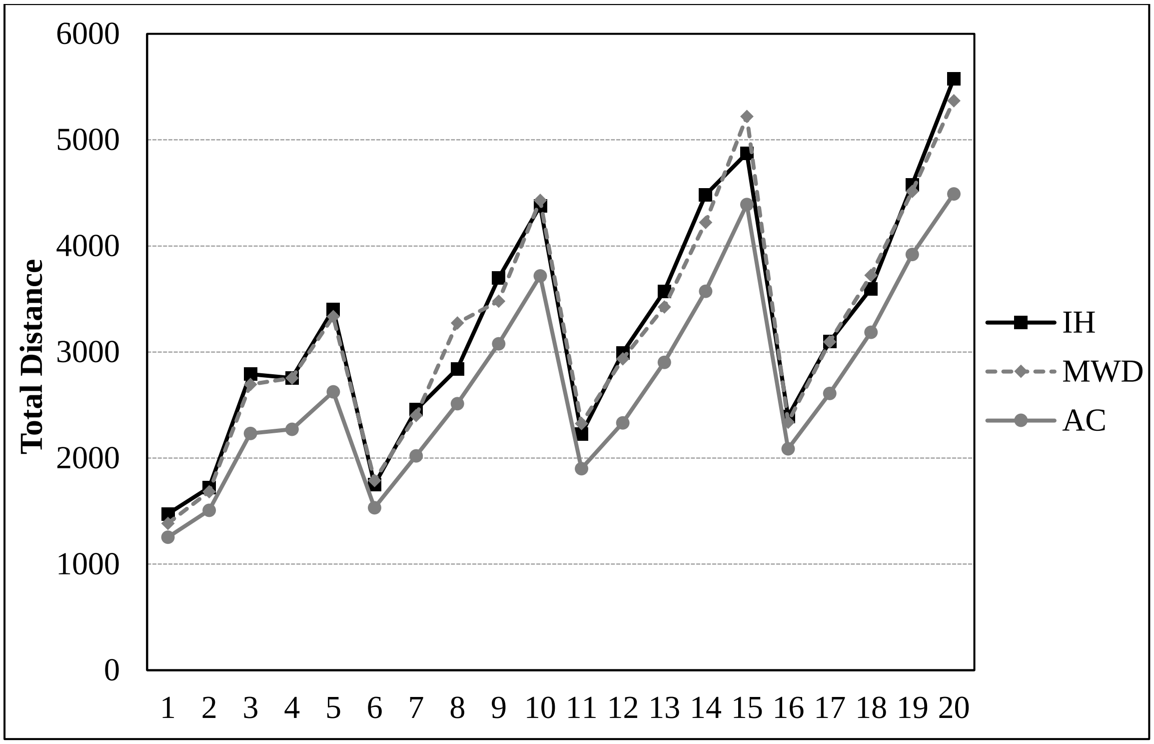

Figure 5.1: The results of the MWD and the IH against the AC algorithm

The RPD of the two heuristics are calculated with respect to the solution found by the AC. The RPD is presented in Table 5.2 and shown in Figure 5.2. The two heuristics have similar performance with an average RPD of 18.92% and 19.40% with respect to the solution found by the AC for the MWD and the IH respectively.

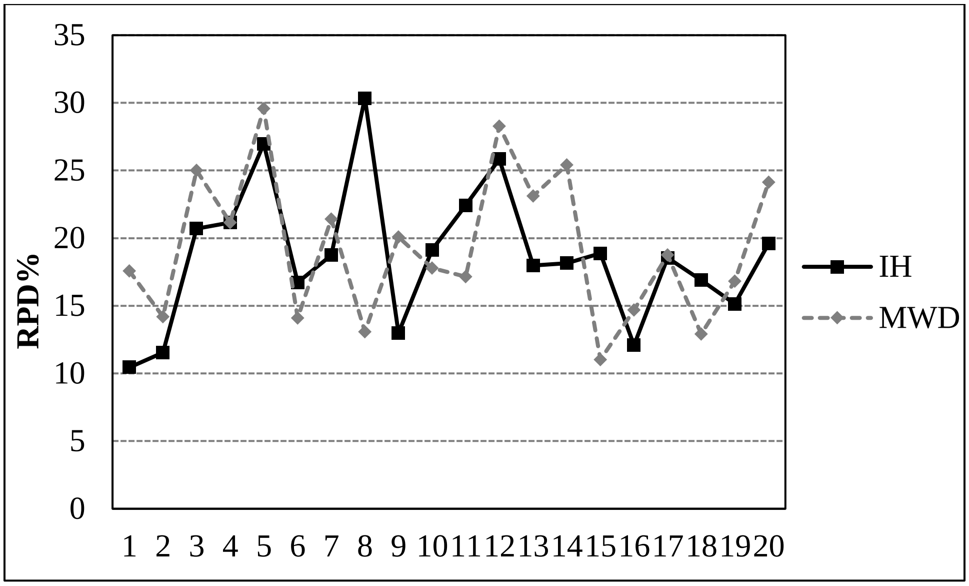

Figure 5.2: The RPD of the IH and the MWD with respect to the AC

5.6 Conclusion

In this chapter, I introduce the VRP arising in the omni-channel retailing. In this problem, the existing retail store distribution system is integrated with the consumer distribution system by utilizing the same fleet of vehicles. The vehicle has two compartments reserved for retail stores demand and consumers demand separately. I present a mathematical model for the problem. The mathematical model was solved using CPLEX to find the optimal solution. I propose different solution approaches; a Minimum weighted distance heuristic, an insertion heuristic and an AC algorithm. Bench mark problems are generated to test the effectiveness of the proposed solution approaches. The AC produced an average relative percentage deviation of 0.10% with respect to the optimal solution for small problem instances. The AC is able to improve the IH and the MWD heuristic solutions by 19% approximately for large problem instances.

This chapter assumes that the consumers are already assigned to retail stores. A suggested research direction is to make the assignment decision as a decision variable in the routing problem. In this case, the retail stores shipping the consumer orders will be decided by the product availability at retail stores. This decision can be made simultaneously while selecting the best routes.

Chapter 6

Vehicle Routing Problem in Omni-Channel Retailing Distribution Systems with Inventory Consideration

6.1 Introduction

Transportation of goods through supply chains has attracted significant interest in the optimization research throughout the last decades. Transportation has been a key element of the competition between organizations as it can denote 10% to 20% of the cost of any product [1]. Operations research has been employed in planning efficient transportation systems by optimizing the distribution of goods. The VRP is an operational decision in the supply chain. The VRP is the problem of distributing goods between depots and customers. In this class of problems, the demands of customers are fulfilled with the products originating from a depot and transported using a fleet of vehicles such that the total traveling cost of all vehicles is minimized. There are many variants of the VRP according to the different classifications of customers and vehicles characteristics. For recent research work in the area of the VRP and its variants refer to [19-23].

The problem considered in this chapter is motivated by the product distribution system found in most retail chain stores following the omni-channel business model. Recently, e-commerce has been a global trend and an important tool for every business worldwide. Economies have started to rely on e-commerce, and companies have been forced to adopt different strategies and policies to adapt to this change in the market. Nowadays, most businesses try to increase their sales by using a recent business model called omni-channel retailing (e.g., John Lewis Partnership, a major retailer in the UK [113]). The retail industry has evolved toward omni-channels [5]. This development is helping retail stores reach more consumers and expand their market. It allows consumers to go shopping seven days a week, twenty-four hours a day. Omni-channel retailing is a seamless experience in which there is no difference between physical and online shopping [90].

In omni-channel retailing, consumers can buy products through multiple channels. They can choose to order products to be delivered at home or to physically visit a store to buy the products. Delivering products ordered online to consumers is the responsibility of the company. These products can be shipped from online distribution centers or from nearby stores depending on product availability. Shipping products from an online distribution center incurs additional cost to the supply chain. It results in the supply chain holding more inventories and delays inventory turnover. In addition, shipping from an online distribution center disrupts the availability between the online system and stores. This means that some products are available at stores but not available online and vice versa. This has been hindering the omni-channel shopping experience and customer satisfaction. Therefore, companies are motivated to have all products available in stores as well as online (i.e., shipping the products ordered online from nearby stores).

Usually, the products ordered online are shipped through regular mail or by a separate fleet of vehicles. Shipping products through regular mail increases the product cost, which is paid either by the consumer or by the company. Similarly, the use of a separate fleet of vehicles to meet the online demand increases product cost as well. However, the additional cost can be reduced if the products ordered online are distributed using the existing distribution system. The model proposed in this chapter is applicable to organizations that use light to medium trucks for serving their scattered retail stores. The proposed model is not applicable to some business models where only large trucks are used to deliver products to retail stores, because it is inappropriate to visit consumers using these large trucks. In addition, several European Union countries have imposed restrictions on the times and days when heavy trucks (gross weight equal to or greater than 7.5 tons) are allowed inside cities for noise and pollution control. In this case, organizations prefer using fleets of smaller trucks to avoid such restrictions. The proposed model is applicable in this situation as well.

The companies that own chain stores operate their own distribution networks to deliver products to different retail stores located in the city. Each distribution network includes a central warehouse, a group of scattered retail stores, and a fleet of vehicles to deliver the products. In this chapter, I propose delivering products ordered online using the same fleet of vehicles. The use of the same vehicles for retail stores and consumers complicates the existing distribution system. However, it reduces the total distribution cost. This chapter addresses the issue of managing this complicated distribution system of the omni-channel retail business model. Thus, I propose a model that uses the same fleet of vehicles to distribute online ordered products and the products required by retail stores. The proposed model integrates two existing distribution systems: the retail distribution system and the consumer distribution system. In the retail distribution system, products are delivered from the depot to the retail stores. In the consumer distribution system, products are delivered from the retail stores to the consumers. Thus, the VRP arising from this integrated distribution system is considered a generalization of two VRP structures. The first VRP structure is the CVRP, and the second VRP structure is the PDP. However, the proposed model is more general compared with the PDP because retail stores can serve more than one consumer. Moreover, the retail stores serving the consumer orders are decided based on product availability, which is not determined in advance in our model.

The proposed problem is considered a generalization of the CVRP and PDP. The problem reduces to the CVRP if the demands of all online orders are considered zero. The problem reduces to the PDP if the demands of all retail stores are considered zero. Both CVRP and PDP are considered NP-hard problems, and hence, the proposed problem will also be NP-hard. To the best of our knowledge, this is the first time the VRP in omni-channel distribution systems is addressed.

The main contribution of this chapter can be stated as follows: A new variant of the VRP arising in omni-channel distribution systems is presented. This problem has practical applications in most retail distribution systems. The mathematical formulation of the described problem is presented. In addition, I present two solution approaches to solve the problem (two-phase heuristic and multi-ant colony (MAC) algorithm). Moreover, new bench mark problem instances are presented that can be used to compare the results of new solution approaches in future research work. Finally, I illustrate the benefit of using the integrated distribution system instead of the two existing distribution systems.

6.2 Literature Review

The proposed problem is closely related to the CVRP and PDP. The CVRP was first considered as a generalized form of the popular TSP and formulated by Dantzig and Ramser [3]. In the last six decades, many exact, heuristic, and meta-heuristic approaches have been proposed to solve the problem. A branch-and-cut algorithm based on a two-commodity network flow formulation was presented by Baldacci et al. [8]. An exact algorithm based on set partitioning formulation was described by Fukasawa et al. [9]. New formulations for the problem and new lower bounds were presented by Letchford and Salazar-Gonzalez [10]. Different types of metaheuristics have been efficiently used to solve the CVRP (e.g., hill climber heuristic (Derigs and Kaiser [11]), particle swarm optimization (Ai and Kachitvichyanukul [12]), memetic algorithm (Nagata and Bräysy [13]), artificial bee colony algorithm (Szeto et al. [14]), and AC algorithm (Reimann et al. [15], and Yu et al. [16])). For recent research in VRP and its variants refer to [19-23].

The PDP was considered for the first time by Lokin [43]. The author introduced a variant of the TSP where precedence relations are forced on some of the customers. This means that some nodes must be visited before other nodes. The author described a branch and bound algorithm for this problem, which was later known as the PDP. Kalantari et al. [44] presented a branch and bound algorithm for the problem. They considered single and multiple capacitated and non-capacitated vehicles. Savelsbergh and Sol [45] presented a survey and a description of the general pickup and delivery problem. Lu and Dessouky [46] presented mixed integer linear programming formulation for the multiple vehicle PDP and proposed a branch-and-cut algorithm. Ting and Liao [47] formulated the selective PDP and presented a memetic algorithm to solve the problem. The PDP with time windows is a generalization of the PDP. Different heuristics and metaheuristics have been used to solve the problem (e.g., a large neighborhood search heuristic (Ropke and Pisinger, [54]), an insertion-based construction heuristic (Lu and Dessouky, [55]), a two-phase heuristic (Lau and Liang, [51]), a reactive TS (Nanry and Barnes [50]), a grouping GA (Pankratz, [53]), a tabu-embedded SA (Li and Lim [52]), a SA, a PSO, a GA, and an artificial immune system (D'Souza et al., [56])). A number of survey papers that deal with the PDP have been recently published. Berbeglia et al. [60] presented a survey and classification for the problem. Parragh et al. [61-62] provided a comprehensive survey and another classification. A literature review shows that the CVRP and PDP have been modeled independently. The problem proposed in this chapter combines the structure of the CVRP and PDP. To the best of our knowledge, this combination has not been presented before.

It is clear from the literature that heuristics and metaheuristics have been used to efficiently solve the CVRP and PDP. The proposed problem is complicated, which forces the solution approach to solve it in two phases. Therefore, the MAC algorithm can be considered as a suitable approach. However, other metaheuristics can be considered in future research and compared with the results reported in this chapter. The AC algorithm was introduced by Dorigo et al. [63]. The algorithm was inspired from the behavior of real ants. Ants communicate by depositing pheromone on every route they travel. Higher pheromone intensity guides the successive ants to the most promising route. The algorithm was first proposed to solve combinatorial problems. It was applied to solve the TSP. Bullnheimer et al. [64] applied the AC algorithm for the first time to solve the VRP with promising results. The AC has been used to efficiently solve different variants of the VRP. It has been used to solve the dynamic VRP [67], the VRP with backhauls [36], the VRP with simultaneous delivery and pickup [37], and the multi-compartment VRP [66]. The MAC algorithm was used to solve the capacitated location routing problem in [41] and the VRP with time windows and uncertain travel times in [17]. It is clear that the AC algorithms have been successfully applied to variants of the VRP with promising results.

6.3 Problem Description and Mathematical Formulation

The problem investigated in this chapter considers a giant retailing company running in different cities. In a given city, it owns a central warehouse and a group of retail stores. The company offers its consumers the choice of purchasing products in two ways: physically visiting the stores or ordering the products online. The proposed model considers the distribution system for this giant retailing company in a particular city only.

The company offers a group of products that can be ordered online. Each consumer places an order for a specific product that can be supplied from more than one retail store. Thus, the consumer demand specifies the type and the quantity of the required product. The available inventory for each product at different retail stores is known. The retail store from which a consumer demand will be satisfied is considered as a decision variable in our model. For simplification, if a consumer orders more than one item, I assume that each item will be delivered separately by duplicating the consumer. The assumption of single item order by a consumer is adopted to guarantee the solution feasibility and to minimize the distribution cost. If multiple items are ordered and can be satisfied from a single retail store, the algorithm will automatically assign single deliveries. If multiple items are ordered and if they are required to be satisfied from multiple retail stores, enforcing single delivery for consumers can either increase the distribution cost or make the solution infeasible.

Products from different suppliers are delivered to the retail company's central warehouse. These products are stored in the warehouse in larger packages till they are needed by the retail stores. I assume that each retail store has a specified delivery from the depot which is measured in terms of number of pallets. Thus, the demand is specified by the number of pallets. The products are delivered to the retail stores on a daily basis in order to replenish their inventory. Moreover, the company commits to delivering the products ordered online from the retail stores to the consumers. Therefore, the distribution network of the company performs two types of deliveries: deliveries from the warehouse to the retail stores and deliveries from the retail stores to the consumers. The company tries to utilize the same fleet of vehicles to serve the retail stores and the consumers.

The demand of the online consumers is satisfied from the retail stores' inventory. Products ordered online are picked up from retail stores and shipped to the online consumers. The products shipped to consumers may be the same products delivered to the retail stores from the warehouse. However, those products need further handling and packaging before being shipped to the consumers. They must be packaged in advance to be picked up by a vehicle visiting the retail stores. Therefore, as they require further packaging, products ordered online cannot be shipped directly from the central warehouse. Moreover, the warehouse does not hold inventory of all items. It temporarily holds the products until they are needed by the retail stores.

The proposed model suggests that both retail stores and consumers are visited using the same fleet of vehicles. Hence, the proposed model is applicable to organizations that use small vehicles for serving their scattered retail stores. The proposed model is not applicable to some business models where only large trucks are used to deliver products to the retail stores because it is inappropriate to visit consumers using these large trucks.

In summary, the proposed model considers two sets of customers served by a fleet of homogenous vehicles. The first set of customers comprises r retail stores that require products to be delivered from a distribution center, and the second set comprises c consumers that require products to be supplied from any retail store. The fleet of vehicles is available at the depot.

A solution for the proposed model comprises a number of vehicle routes. Each vehicle starts from the warehouse, visits a set of retail stores and consumers in any sequence, and finally returns to the warehouse. However, the retail store that has been determined to satisfy the demand of a particular consumer must be visited before that particular consumer. The proposed model aims to minimize the total cost of the vehicle routes such that:

- Every route starts and ends at the depot.
- The routes do not exceed the maximum tour length.
- Every retail store is visited by only one vehicle and only once.
- The total load of the vehicle arising from the delivery demand of the retail stores does not exceed the vehicle capacity.
- Every consumer is visited by only one vehicle and only once.
- The total consumer demand to be fulfilled by a retail store for a certain product does not exceed the available inventory of this product at the retail store.
- The retail store determined to satisfy the consumer’s demand must be visited before the consumer and by the same vehicle.

The problem structure is illustrated using a numerical example. Consider a small distribution network that consists of one depot and three retail stores and six consumers. Three products are available for online orders and the vehicles capacity is 100. The data of the distribution network is given in Table 6.1:

Table 6.1: Data for a small distribution network

Node	X	Y	D_i	Ordered product	Online demand	Inventory		
						P1	P 2	P 3
0	95	66	0	0	0	0	0	0
1	54	23	37	0	0	0	2	1
2	19	40	42	0	0	1	2	0
3	75	19	28	0	0	2	0	1
4	60	20	0	3	1	0	0	0
5	8	89	0	2	1	0	0	0
6	35	42	0	2	1	0	0	0
7	60	10	0	1	1	0	0	0
8	48	93	0	1	1	0	0	0
9	41	99	0	2	1	0	0	0

The solution of the proposed problem specifies two decisions: the assignment of consumers to retail stores and the vehicle route. The optimum solution of the problem has a total distance of 386.91 and the vehicle route is (0 1 6 2 5 9 8 0 3 7 4 0). In this solution, as shown in Figure 6.1, consumer 6 is assigned to retail store 1; consumers 5, 8, and 9 are assigned to retail store 2; and consumers 4 and 7 are assigned to retail store 3. The first vehicle starts from the depot (node 0) with initial load of 79 units to satisfy the demand of retail stores 1 and 2. The vehicle then proceeds to retail store 1 to deliver its demand (37 units) and to pick up the product required by its assigned consumer 6. The vehicle delivers the picked-up product to consumer 6. The vehicle then proceeds to retail store 2 to deliver its demand (42 units) and to pick up the product required by its assigned consumers 5, 9, and 8. The vehicle then delivers the picked-up product to consumers 5, 9, and 8 and returns to the depot. The second vehicle starts from the depot (node 0) with initial load of 28 units to satisfy the demand of retail store 3. The vehicle then proceeds to retail store 3 to deliver its demand and to pick up the product required by assigned consumers 7 and 4. The vehicle delivers the picked-up product to consumers 7 and 4 and returns back to the depot.

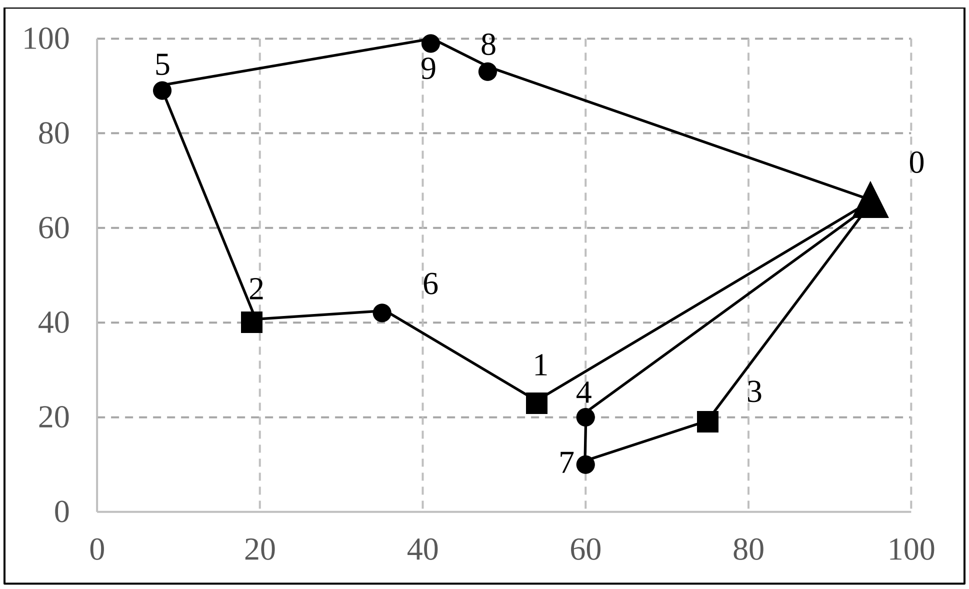

Figure 6.1: The optimum solution of the distribution network in Table 6.1

The notations used in the mathematical formulation are listed below:

Sets:

N Set of all customer nodes

R Set of retail stores

C Set of online consumers (consumers ordering products online)

K Set of vehicles

P Set of products

Parameters:

Q Vehicle capacity

D_i Quantity to be delivered from the depot to retail store *i*

d_j^p Demand of consumer *j* for product *p*

I_i^p Inventory level of product *p* at retail store *i*

L The maximum length of any route

C_{ij} The distance of traversing arc (*i*,*j*)

T_{ij} The time of traversing arc (*i*,*j*)

O_i Drop time at the location *i*

Decision variables:

Q_i^k The total quantity carried by vehicle *k* after leaving node *i*

S_i^k The service start time at node i by vehicle k

X_{ij}^k Binary flow variable equals 1 if the arc (*i*,*j*) is traversed by vehicle *k*, and 0 otherwise.

Y_{ij} Binary variable equals 1 if consumer $j \in C$ is served by retail store $i \in R$ and 0 otherwise.

The problem can be defined using graph theory as follows: Let $G = (V, A)$ be an undirected graph with a set of vertices $V = \{0, 1, \ldots, (r+c)\}$, where r is the number of retail stores and c is the number of consumers. Node {0} is the distribution center from where products are delivered to retail stores. Here, R = {1 … r} are the retail stores, and C = {r+1… r+c} are the consumers. Nodes N = {1, 2, …r, r+1, …, r+c} are the customer nodes (retail stores and consumers) served by a number of identical vehicles k (initially located in the depot). There are P types of products available for orders by consumers. Each customer $i \in N$ has a quantity D_i to be delivered from the depot. According to the definition, D_i is the quantity to be delivered from the depot to retail store i. Since the demand of online consumers is not satisfied from the depot, D_i is considered to be zero for all online consumers. Each retail store $i \in R$ has an inventory level I_i^p available for satisfying orders of consumers for a certain product $p \in P$. Each consumer $j \in C$ has a demand d_j^p for product p, which can be delivered from any retail store $i \in R$. However, I assume that each consumer can order only one item. I assume that the consumer demand is negligible compared to the retail store demand. Therefore, the vehicle load is not affected by the consumer demand. The retail store from where the consumer order is shipped is decided according to inventory availability.

$$Z = \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} C_{ij}^k X_{ij}^k \tag{6.1}$$

Subject to:

$$\sum_{k \in K} \sum_{j \in V} X_{ij}^k = 1 \qquad \forall i \in N, i \neq j \tag{6.2}$$

$$\sum_{j \in R} X_{(0)j}^k = 1 \qquad \forall k \in K \tag{6.3}$$

$$\sum_{i \in V} X_{ij}^k - \sum_{i \in V} X_{ji}^k = 0 \qquad \forall j \in N, k \in K, i \neq j \tag{6.4}$$

$$\sum_{i \in N} X_{i(0)}^k = 1 \qquad \forall k \in K \tag{6.5}$$

$$Y_{ij} \left(\sum_{l \in N} X_{il}^k - \sum_{l \in N} X_{lj}^k\right) = 0 \qquad \forall i \in R, j \in C, k \in K, p \in P, i \neq l, j \neq l \tag{6.6}$$

$$\sum_{i \in R} Y_{ij} = 1 \qquad \forall j \in C, p \in P \tag{6.7}$$

$$\sum_{j \in C} Y_{ij} d_j^p \leq I_i^p \qquad \forall i \in R, p \in P \tag{6.8}$$

$$Q \geq Q_i^k \geq (Q_j^k + D_j) X_{ij}^k \qquad \forall i \in V, j \in N, k \in K, i \neq j \tag{6.9}$$

$$S_j^k \geq (S_i^k + T_{ij} + O_i)\, X_{ij}^k \qquad \forall i \in V, j \in N, k \in K, i \neq j \tag{6.10}$$

$$S_j^k \geq (S_i^k + T_{ij} + O_i)\, Y_{ij} \qquad \forall i \in R, j \in C, k \in K \tag{6.11}$$

$$S_i^k + O_i + T_{i0} \leq L \qquad \forall i \in N, k \in K \tag{6.12}$$

$$X_{ij}^k \in \{0,1\} \qquad \forall i \in V, j \in V, i \neq j, k \in K \tag{6.13}$$

$$Y_{ij} \in \{0,1\} \qquad \forall i \in R, j \in C, p \in P \tag{6.14}$$

Eq. (6.1) is the objective function representing the total traveling cost of all arcs traversed by all vehicles. Eq. (6.2) confirms that exactly one arc enters a customer node. Eqs. (6.3-6.5) confirm that each vehicle starts its route from the depot and ends its route at the depot. Eq. (6.6) confirms that if a retail store is fulfilling the demand of a certain consumer, both the retail store and consumer are visited by the same vehicle. Eq. (6.7) confirms that each consumer is served by only one retail store. Eq. (6.8) confirms that the quantity to be delivered from retail store *i* to all the consumers served from this retail store is less than or equal to the inventory level at this retail store. Eqs. (6.9) and (6.10) confirm that the consistency of the capacity and time and the routes connectivity are preserved. Eq.

(6.11) confirms that for each delivery, the retail store i is visited before consumer j. Eq. (6.12) confirms that no route should exceed the maximum tour length.

The mathematical model is nonlinear and the problem falls under the category of NP-hard problems. However, the mathematical model is linearized and solved using CPLEX to obtain optimal solutions for small problem instances. The optimal solutions found by CPLEX are used to evaluate the performance of the proposed solution approaches. The model is linearized by replacing Eq. (6.6) with Eqs. (6.15 and 6.16) and replacing Eqs. (6.9-6.11) with Eqs. (6.17-6.19) respectively.

$$\sum_{l \in N} X_{il}^{k} - \sum_{l \in N} X_{lj}^{k} \leq (1 - Y_{ij})M \qquad \forall i \in R, j \in C, k \in K \qquad (6.15)$$

$$\sum_{l \in N} X_{lj}^{k} - \sum_{l \in N} X_{il}^{k} \leq (1 - Y_{ij})M \qquad \forall i \in R, j \in C, k \in K \qquad (6.16)$$

$$(Q_j^k + D_j - Q_i^k) \leq (1 - X_{ij}^k)M \qquad \forall i \in V, j \in N, k \in K, i \neq j \qquad (6.17)$$

$$(S_i^k + T_{ij} + DT_i - S_j^k) \leq (1 - X_{ij}^k)M \qquad \forall i \in V, j \in N, k \in K, i \neq j \qquad (6.18)$$

$$(S_i^k + T_{ij} + DT_i - S_j^k) \leq (1 - Y_{ij})M \qquad \forall i \in R, j \in C, k \in K \qquad (6.19)$$

Where *M* is a number with a big value.

6.4 Solution Approaches

It is common in the VRP literature to consider the near-optimum solutions found using heuristic and metaheuristic solution approaches. Thus, this chapter presents a two-phase heuristic and a metaheuristic based on the AC algorithm to solve the problem. Heuristics are used when reasonable quality solutions are required in minimal CPU time. Metaheuristics can generate better quality solutions but require longer CPU time. This chapter aims to provide both solution approaches that can satisfy the need of different users.

6.4.1 Two-Phase-Based Heuristic

The heuristic was designed in two phases to ensure that it builds a feasible route. In the first phase, consumers are assigned to different retail stores and simultaneously a retail store route is built for each retail store. In the second phase, retail stores' routes are combined to build the final solution. The proposed problem assumes that the tour length of a given route does not exceed a specified tour length. The problem also requires that the demand of consumers is satisfied from the product available at the retail stores. Further, the problem assumes a single visit for each retail store and consumer. This means that a retail store and its associated consumers must be visited by the same vehicle. Then, in a simplest route, the vehicle starts from a depot, visits a single retail store and its associated consumers, and finally returns to the depot. In this situation, it is necessary to assign consumers to retail stores such that the simplest route consisting of a retail store and its associated consumers does not violate the maximum tour length constraint. Thus, our heuristic approach tries to find a feasible assignment of consumers to retail stores in the first phase. Then, retail stores' routes are combined in the second phase to reduce the total tour distance. The heuristic is described below in more details.

6.4.1.1 Phase 1: Build TSP Routes

The aim of the first phase of the heuristic is to build feasible retail store routes. In the beginning, routes equal to the number of retail stores are built. The route starts from a depot, proceeds to one of the retail stores, and returns to the depot. Then, consumers are assigned to retail stores' routes starting from the first consumer till all consumers are assigned to the routes. First, efforts are made to assign each consumer to its nearest retail store on the basis of two criteria. The first criterion is that the product required by the consumer is available at the retail store. The second criterion is that the insertion of the consumer in the least cost position on the retail store's route does not violate the maximum tour length constraint. The least cost position is the position at which the length of the retail store's route is the minimum. If the consumer assignment at the nearest retail store is not feasible, then an attempt is made to assign the consumer to the next nearest route. In this manner, the consumer is assigned to one of the retail stores' routes. If the consumer cannot be assigned to any retail store route without exceeding the maximum tour length, then a corrective action is used for consumer assignment.

In the corrective action, the consumer is assigned to its nearest retail store where inventory is available to satisfy the consumer demand. This is done by removing one or more existing consumers from the retail store's route. The removal of existing consumers starts from the last consumer in the route. The removed consumers are assigned to other retail stores' routes. The existing consumers can be removed from their assigned retail store's route only if they can be feasibly assigned to another route. I keep on removing existing consumers till the new consumer can be inserted without violating the maximum tour length constraint. If removing existing consumers from the retail store's route does not allow the new consumer to be feasibly inserted, an attempt is made to insert the new consumer in other retail stores' routes. Priority of insertion is given to the nearest retail store.

6.4.1.2 Phase 2: Build Final VRP Routes

In the second phase of the heuristic, the routes of the retail stores are combined. Two retail stores' routes can be combined only if it is possible to complete the two routes and return to the depot without violating the maximum tour length. The routes are combined based on the maximum saving criterion of Clarke and Wright [114]. Under this criterion, combinations of all existing routes are considered for possible merging. The combination that provides the maximum benefit is selected for merging. The merging process continues till merging of two routes is feasible.

6.4.2 MAC Algorithm

The power of the AC and other metaheuristics comes from their ability to approach good quality solutions in minimal time. They start with a group of initial solutions scattered in the search space, and then proceed to the area where it is most likely to find better solutions. In addition, metaheuristics have the ability to avoid trapping in the local optimum unlike regular heuristics. The probability of choosing lower quality solutions enables the heuristic to search new areas where good quality solutions may be hidden. The proposed problem is complicated, which forces the solution approach to solve it in two phases. Therefore, the MAC algorithm can be considered as a suitable approach unlike some of the other metaheuristics. However, other metaheuristics can be considered in future research and then compared with the reported results in this chapter. The AC algorithm is based on the behavior of real ants when they search for the shortest path to food. Real ants follow the trails of their successive ants to find the shortest path. The AC uses artificial ants to construct good quality solutions. In the MAC algorithm, more than one ant is used to construct a solution. In our MAC algorithm, the first group of ants (Ant_1) is used to assign consumers to retail stores and build retail stores' routes. The second group of ants (Ant_2) is used to build final vehicles routes by combining retail stores' routes produced by the first ant. The outline of the algorithm is given below followed by a detailed description:

Step 1: Create initial solution to initialize the trail intensities

Step 2: Repeat the following to construct VRP routes

- Use Ant_1 to build retail stores' routes
- Use Ant_2 to build VRP routes by combining retail stores' routes
- Perform local search to improve the solution quality of ants
- Update best solution found so far
- Update trail intensities for all arcs using best solution

Step 3: Terminate the algorithm and report the best solution

6.4.2.1 Initializing Trail Intensities

The first step of the MAC algorithm is to initialize trail intensities (pheromone) to all arcs. This initial pheromone is usually calculated from an initial solution. In this algorithm, the initial pheromone is calculated from the solution found in the two-phase heuristic. The initial pheromone does not affect the solution because the same amount of pheromone is located on all edges. The initial pheromone is calculated based on the expression $\tau 1_{ij} = \tau 2_{ij} = 1/L \quad (\forall i, j \in N)$, where L is the total length of the route generated using the two-phase heuristic.

6.4.2.2 Route Construction

A number of *m* routes are generated in each iteration of the MAC using *m* ants from Ant_1 and *m* ants from Ant_2.

1. Building Retail Stores Routes Using Ant1

Ant_1 is used to assign consumers to retail stores, i.e., generate a route for each retail store. The ant assigns consumers one by one to retail stores based on the probability function stated in Eq. (6.20), where $N1_j$ is the list of feasible retail stores that have sufficient inventory of the product required by consumer i. The term $\varepsilon 1_{ij}$ is the attraction value of assigning consumer *i* to retail store *j* and is calculated using Eq. (6.20). The first measure $\tau 1_{ij}$ is the pheromone value between retail store *j* and consumer *i*. This value represents how frequently consumer *i* was assigned to retail store *j* in previous iterations. The second measure $\mu 1_{ij}$ is the inverse of the arc *(i,j)* distance between retail store *j* and consumer *i*. The lower distance gives the higher attraction value. The third measure ST_j is the slack time in the retail store *j* trip (i.e., the difference between the maximum tour length and the current tour length of retail store *j*). It can be noticed that it is more preferable to assign consumers in the routes with more slack time. However, if the trip time exceeds the maximum tour length, the slack time is equal to 1. This means that consumers can still be assigned to retail stores even if it violates the maximum tour length constraint. The algorithm is allowed to exceed the maximum tour length in order to offer more chances to escape trapping in the local optima. However, after all retail stores' routes are generated, a corrective action is performed to meet the maximum tour length constraint.

$$P1_{ij} = \begin{cases} \dfrac{\varepsilon 1_{ij}}{\sum_{l \in N1_i} \varepsilon 1_{li}} & if\ j \in N1_i \\ 0 & otherwise \end{cases} \tag{6.20}$$

$$\varepsilon 1_{ij} = \left(\tau 1_{ij}\right)^{\alpha 1} \left(\mu 1_{ij}\right)^{\beta 1} \left(ST_j\right) \tag{6.21}$$

After assigning each consumer to a retail store, all possible positions in the retail stores' route are considered for inserting the consumer. The consumer is inserted at the position that generates a shorter distance route. Ant_1 ignores the maximum tour length constraint by allowing assignment of consumers to routes even if there is no slack time in these routes. Thus, a corrective action must be performed to make the routes feasible.

A corrective action is applied to the retail stores' routes violating the maximum tour length constraint. Starting from the last consumer, consumers are considered for removal from the infeasible route one by one till the route becomes feasible. The removed consumers are assigned to other retail stores where they can be feasibly assigned. If a consumer cannot be feasibly assigned to another retail store's route, it is not removed from its original route. If the infeasible routes remain infeasible after possible removal of all existing consumers, the entire solution is discarded and another solution is generated.

2. Combining Retail Store Routes Using Ant2

Ant_2 is used to combine retail stores' routes, i.e., generate final VRP routes. The probability of combining routes of retail store *j* and retail store *i* is calculated using Eq. (6.22), where $N2_i$ is the list of feasible retail stores' routes that can be combined with the route of retail store *i* without exceeding the maximum tour length constraint. The term $\varepsilon 2_{ij}$ is the attraction value of combining the routes of retail store *i* and retail store *j* and is calculated using Eq. (6.23). The first measure $\tau 2_{ij}$ is the pheromone value between retail store *i* and retail store *j*. This value represents how frequently the routes of retail store *i* and retail store *j* are combined in the best solutions found in the previous iterations. The second measure $\mu 2_{kj}$ is the inverse of arc (*k*,*j*) distance between retail store *j* and consumer *k*, where *k* is the last consumer in the route of retail store *i*. The lower arc distance gives the higher attraction value.

$$P2_{ij} = \begin{cases} \dfrac{\varepsilon 2_{ij}}{\sum_{l \in N2_i} \varepsilon 2_{il}} & if\ j \in N2_i \\ 0 & otherwise \end{cases} \tag{6.22}$$

$$\varepsilon 2_{ij} = \left(\tau 2_{ij}\right)^{\alpha 2} \left(\mu 2_{kj}\right)^{\beta 2} \tag{6.23}$$

By combining retail stores' routes, *m* VRP routes are generated. The distances of all VRP routes are calculated and the best solution found so far is updated.

After creating vehicle routes, insertion local search is used to improve the solution quality. Two neighborhood structures are investigated: intra-route move and inter-route move. Feasible moves are evaluated and the move with highest improvement is applied. The local search is stopped when no further improvement is allowed.

6.4.2.3 Updating Elitist Ants and Trail Intensities

Elitist ants are used to update trail intensities. Elitist ants are defined as the best ant solutions found so far. They are updated by comparing the current ant solution with the current elitist ant solutions. After the elitist ants are updated, all trail intensities are updated using the γ elitist ants found so far. Pheromone or trail intensities evaporate with time on all arcs. At the same time, ants deposit pheromone on the visited arcs. Updating trail intensities is described in two steps: lowering pheromone on all arcs and pheromone increase on the arcs reported in the γ elitist ant routes only. According to Dorigo et al. [63], ρ is defined as trail persistence $0 \leq \rho < 1$. The term $(1-\rho)$ is interpreted as trail evaporation. There are two types of pheromones: pheromone used for assigning consumers to retail stores and pheromone for combining retail stores. Eq. (6.24) is used to update the pheromone used for assigning consumers to retail stores. Pheromone is updated by adding $1/L^{best}$ to the remaining amount of trail intensity after its evaporation. The pheromone is increased if consumer i is assigned to retail store j in the elitist ant solutions. Eq. (6.26) is used to update the trail between the retail stores. Pheromone is updated by adding $1/L^{best}$ to the remaining amount of trail intensity after its evaporation. The pheromone is increased if the routes of retail store i and retail store j are combined in the γ elitist ant solutions, where L^{best} is the total length of the best solution found so far.

$$\tau 1_{ij}^{new} = \rho 1 \times \tau 1_{ij}^{old} + \sum_{\theta=1}^{\gamma} \Delta\tau 1_{ij}^{t\theta} \tag{6.24}$$

$$\Delta\tau 1_{ij}^{\theta} = \begin{cases} 1/L^{best} & \text{if consumer } i \text{ is assigned to retail store} j \\ & \text{in the route of the } \theta\text{th elitist ant,} \\ 0 & \text{otherwise} \end{cases} \tag{6.25}$$

$$\tau 2_{ij}^{new} = \rho 2 \times \tau 2_{ij}^{old} + \sum_{\theta=1}^{\gamma} \Delta\tau 2_{ij}^{t\theta} \tag{6.26}$$

$$\Delta\tau 2_{ij}^{\theta} = \begin{cases} 1/L^{best} & \text{if retail stores } i \text{ and } j \text{ are combined} \\ & \text{in the route of the } \theta\text{th elitist ant,} \\ 0 & \text{otherwise} \end{cases} \tag{6.27}$$

Refer to parameter setting in the next subsection for the setting of the value of ρ. The first parts of Eqs. (6.24) and (6.26) represent the remaining amount of trail intensities after evaporation. Therefore, the existing trail intensities are multiplied by the term ρ. In the second part of Eq. (6.24), trail intensities between consumers and their assigned retail stores in the best solution are increased by the amount $1/L^{best}$. In the second part of Eq. (6.26), trail intensities between retail stores combined in one route in the best solution are increased by the amount $1/L^{best}$.

6.4.2.4 Parameters Setting for MAC

I use 1000 iterations to test the performance of the MAC against the solution found using the two-phase heuristic. Using 1000 iterations kept the computational time of the MAC comparable to the computational time of the two-phase heuristic. Moreover, the solution quality does not significantly improve when more iterations are used. The number of used ants, *m*, controls the solution quality, but it also affects the computational time. I found that after 10 ants, there is no significant improvement in the solution quality compared to the excessive increase in the computational time. I found that using 10 elitist ants to update the trail intensities gives better solution quality. Dorigo et al. [63] reported that better solutions are found using $\alpha 1 = \alpha 2 = 1$ and $\beta 1 = \beta 2 = 2$. Setting $\rho 1 = \rho 1 = 0.9$ updates the pheromone with new ant experience on account of the existing experience, which produces better solutions.

6.5 Numerical Experiments

In this section, I explain the data generation of bench mark problem instances, show and compare the results for the proposed solution approaches, and illustrate the benefit of using the integrated distribution system instead of the two existing distribution systems.

6.5.1 Data Generation

There are no bench mark problem instances for VRP in omni-channel retail distribution systems. Therefore, I create bench mark problem instances to test our solution approaches. I use two sets of problem instances: small problem instances and large problem instances. The small problem instances are used to compare the solution of the proposed solution approaches with the optimal solution. The large problem instances are used to compare the two solution approaches and for testing other solution approaches in the future. In the first set, I use number of retail stores of 3, 4, 5, and 6 and number of consumers of 6, 9, 12, 15, and 18. Thus, a total of 20 problem instances are generated. In the second set, I use number of retail stores of 10, 15, 20, and 25 and number of consumers of 25, 50, 75, 100, and 150, and three different available inventory scenarios. Thus, a total of 60 problem instances are generated. In both sets of problem instances, the X and Y coordinates of retail stores and consumers are randomly created in the range [0,100]. The demand of the retail stores is generated randomly between 25 and 50 units, while the capacity of the vehicles is fixed to 100 units for all problem instances. The number of product types available for online demand generated in [3–5] product types in the first set of problem instances and in [10–20] product types in the second set of problem instances. Each consumer is allowed to choose one item from the products available in the network. Products are randomly assigned to consumers. The maximum tour length is fixed to 8 h and the drop time is fixed to 5 min for all problem instances. Three scenarios for the available inventory of retail stores used to serve the online order are considered to study the effect of inventory availability on the solution in the second set of problem instances. The three scenarios are presented below:

1. **Scenario 1: Tight Available Inventory**
   - $\sum I_p = \sum D_p + U[0.1,0.2] \sum D_p$
2. **Scenario 2: Relaxed Available Inventory**
   - $\sum I_p = \sum D_p + U[0.5,1] \sum D_p$
3. **Scenario 3: Abundant Available Inventory**
   - $I_p = \sum D_p$ (at each retail store)

where

$\sum I_p$ is the total inventory available in the network of product *p*

$\sum D_p$ is the total online demand of product *p*

The first scenario represents tight available inventory to satisfy the consumer demand. The second scenario represents relaxed available inventory and the third scenario represents abundant available inventory. In the first scenario, the total available inventory for a product is just 10%–20% excess of the total product demand. While in the second scenario, the total available inventory for a product is 50%–100% excess of the total product demand. In the third scenario, it is assumed that each retail store can satisfy the demand of all online orders for a certain product from its available inventory.

6.5.2 Computational Results

The proposed MAC algorithm and two-phase heuristic were coded in C and the generated 80 problems were solved. The MAC was run for 1000 iterations only to maintain the considerably low computational time. The two-phase heuristic gave a single solution.

The results were obtained using a server that operates four 2.1 GHz processors with 16-cores each and a total of 256 GB RAM. The mathematical model was coded using AMPL and solved using the CPLEX solver. The performances of the two solution approaches are evaluated with respect to their solutions' total traveled length using the *PD*. The *PD* is calculated according to Eq. (6.28), where PD_i represents the relative percentage deviation in the total length for the two solutions of problem *i*. Furthermore, the term C_i is the total length found by CPLEX for problem *i* and the term S_i is the total length found by the MAC algorithm or the two-phase heuristic for the same problem *i*.

$$PD_i = \frac{S_i - C_i}{C_i} \times 100\% \qquad (6.28)$$

The results of the small problem instances are presented in Table 6.2. The CPU time taken by the MAC algorithm and the two-phase heuristic was less than one second. Note that the MAC algorithm could reach the best solution found by the CPLEX in 17 out of the 20 problem instances. The average PD of the MAC with respect to the solution found by CPLEX is 0.13%. It can be concluded that the MAC algorithm has a good performance while it succeeded in maintaining a shorter CPU time compared with the CPLEX, which needed 4161 s on average. On the other hand, the performance of the two-phase heuristic is poor and it has an average PD of 43% with respect to the solution found by the CPLEX.

Table 6.2: The results of the MAC and the heuristic against the optimum solution.

Problem	Number of Retail stores	Number of Consumers	CPLEX		MAC		Heuristic	
			Total Distance	CPU Time	Total Distance	PD%	Total Distance	PD%
1	3	6	386.91	1	386.91	0.00	479.19	23.85
2	3	9	416.35	1	416.35	0.00	589.63	41.62
3	3	12	424.31	9	424.31	0.00	512.32	20.74
4	3	15	455.29	31	455.29	0.00	767.06	68.48
5	3	18	601.36	65	601.36	0.00	936.20	55.68
6	4	6	419.32	1	419.33	0.00	512.12	22.13
7	4	9	455.91	16	455.91	0.00	688.39	50.99
8	4	12	448.83	525	449.35	0.12	711.88	58.61
9	4	15	457.32	3	457.32	0.00	746.57	63.25
10	4	18	514.38	11326	514.38	0.00	740.80	44.02
11	5	6	486.04	9	486.04	0.00	545.72	12.28
12	5	9	624.81	35	624.81	0.00	881.95	41.15
13	5	12	535.44	33	535.44	0.00	961.67	79.61
14	5	15	605.03	111	605.03	0.00	838.93	38.66
15	5	18	708.23	3078	709.89	0.23	898.75	26.90
16	6	6	468.89	7	468.89	0.00	582.10	24.14
17	6	9	468.61	75	468.61	0.00	608.99	29.96
18	6	12	586.62	6696	586.62	0.00	924.41	57.58

19	6	15	750.94	940	750.94	0.00	964.11	28.39
20	6	18	588.65	60267	601.71	2.22	1005.72	70.85
Average			520.16	4161.45	520.92	0.13	744.82	42.94

The results of the proposed MAC algorithm and the two-phase heuristic for large problem instances are presented in Tables 6.3–6.5. The performances of the two solution approaches are evaluated with respect to the best solution found using the RPD. The RPD is calculated according to Eq. (6.29), where RPD$_i$ represents the RPD in the total length of problem i. Furthermore, the term B_i is the best total length for problem *i*. The RPD presented in Tables 6.3–6.5 are calculated for the two-phase heuristic with respect to the MAC algorithm because the MAC algorithm gave the best solution in all problems. The term S_i is the total length found by the two-phase heuristic for the same problem i.

$$PD_i = \frac{S_i - B_i}{B_i} \times 100\% \tag{6.29}$$

The results for scenario 1 (the tight available inventory scenario) are reported in Table 6.3. It can be noticed that the results obtained from the MAC are better than the results obtained from the two-phase heuristic. The average total length calculated using the heuristic is 3275.61 and the average total length calculated using the MAC is 2028.30. The average RPD of the heuristic compared to the MAC is 61.50%. The CPU time taken for the two-phase heuristic is still less than one sec. However, the MAC CPU time is 227 s on average. The results for scenarios 2 and 3 are reported in Table 6.4 and Table 6.5, respectively. Over the 60 problem instances, the average RPD of the heuristic compared to the MAC is 53.32% and the average CPU time of the MAC is 697 sec.

Table 6.3: The results of inventory scenario 1 using the MAC and the heuristic

Problem	No. of retail stores	No. of consumers	Heuristic Total Length	MAC Total Length	MAC CPU Time	RPD%
1	10	25	1631.63	1002.47	7	62.76
2	10	50	2057.50	1189.79	47	72.93
3	10	75	3006.19	1815.42	80	65.59
4	10	100	2830.16	1529.04	303	85.09
5	10	150	3478.72	1905.19	640	82.59
6	15	25	1774.35	1313.69	8	35.07
7	15	50	2461.83	1510.55	49	62.98
8	15	75	3545.11	2101.77	142	68.67
9	15	100	3528.98	2329.53	227	51.49
10	15	150	4916.75	3012.18	469	63.23
11	20	25	2432.56	1611.34	12	50.97
12	20	50	2695.34	1800.87	55	49.67
13	20	75	3936.67	2406.04	140	63.62
14	20	100	3826.14	2483.81	345	54.04
15	20	150	4496.05	2790.69	761	61.11

16	25	25	2254.87	1669.56	14	35.06
17	25	50	3020.80	1965.56	50	53.69
18	25	75	3963.52	2449.76	146	61.79
19	25	100	4933.86	2788.48	270	76.94
20	25	150	4721.26	2890.29	762	63.35
Average			3275.61	2028.30	226.35	61.50

Figure 6.2 shows the effect of the number of retail stores on the heuristic performance. It can be noticed from Figure 6.2 that the RPD of the heuristic with respect to the MAC decreases as the number of retail stores increases. In other words, the performance of the heuristic increases as the number of retail stores increases. In the case of a smaller number of retail stores, the problem is harder owing to the maximum tour length constraint. In this case, the performance of the MAC is better than that of the heuristic.

It can be noticed from Figure 6.3 that, the overall RPD of the heuristic with respect to the MAC increases as the number of consumers increases. In other words, the performance of the MAC increases as the number of retail stores increases. In the case of a higher number of consumers, the problem is harder and the MAC can perform better than the heuristic.

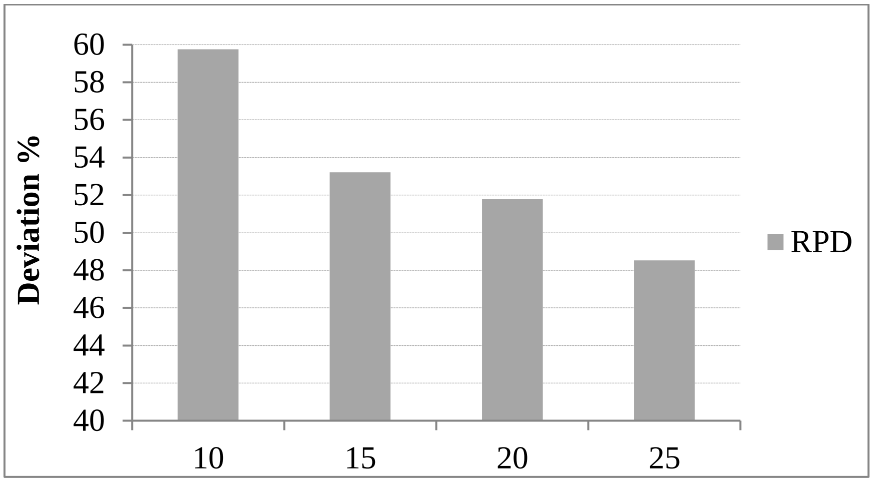

Figure 6.2: The effect of the number of retail stores on the RPD

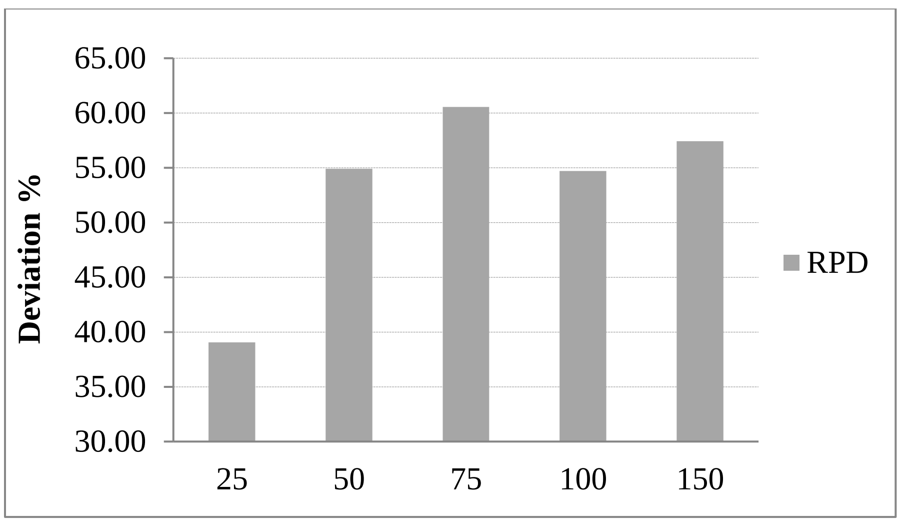

Figure 6.3: The effect of the number of consumers on the RPD

6.5.2.1 Effect of the Available Inventory on the Performance of the Solution Approaches

From Tables 6.3–6.5, the averages of the total lengths calculated using the heuristic are 3275.61, 2812.91, and 1824.49 for the available inventory scenarios 1, 2, and 3, respectively. The averages of the total lengths calculated using the MAC are 2028.30, 1716.91, and 1346.20 for the available inventory scenarios 1, 2, and 3, respectively. The average RPD of the heuristic compared to the MAC is 61.50%, 63.84%, and 35.53% for the available inventory scenarios 1, 2, and 3, respectively. The RPD of the two-phase heuristic compared to the MAC decreases when the more relaxed available inventory scenario is used. This means that the two-phase heuristic performance increases when the problem is more relaxed. In other words, the MAC can still search and find good quality solutions efficiently even if the problem is tighter. This is shown in Figure 6.4.

Table 6.4: The results of inventory scenario 2 using the MAC and the heuristic

Problem	No. of retail stores	No. of consumers	Heuristic Total Length	MAC Total Length	MAC CPU Time	RPD%
21	10	25	1571.63	879.16	11	78.76
22	10	50	1920.59	1083.66	92	77.23
23	10	75	2699.23	1591.52	180	69.60
24	10	100	2305.09	1437.68	569	60.33
25	10	150	2700.41	1519.35	1979	77.73
26	15	25	1665.15	1180.83	12	41.02
27	15	50	2320.66	1329.26	81	74.58
28	15	75	3016.45	1692.41	305	78.23
29	15	100	3302.39	2016.40	605	63.78
30	15	150	3918.97	2399.60	1547	63.32

31	20	25	1993.49	1495.79	17	33.27
32	20	50	2713.01	1656.86	82	63.74
33	20	75	3393.34	1799.64	277	88.56
34	20	100	3127.49	2018.50	798	54.94
35	20	150	3742.21	2205.53	2364	69.67
36	25	25	2032.06	1530.33	17	32.79
37	25	50	3130.52	1939.50	80	61.41
38	25	75	3433.23	2088.58	302	64.38
39	25	100	3824.46	2244.14	700	70.42
40	25	150	3447.89	2229.43	2041	54.65
Average			2812.91	1716.91	602.95	63.84

Table 6.5: The results of inventory scenario 3 using the MAC and the heuristic

Problem	No. of retail stores	No. of consumers	Heuristic Total Length	MAC Total Length	MAC CPU Time	RPD%
41	10	25	897.55	710.47	17	26.33
42	10	50	1287.81	875.42	154	47.11
43	10	75	1531.06	1114.87	384	37.33
44	10	100	1636.54	1247.46	1000	31.19
45	10	150	1551.75	1273.86	2158	21.81

46	15	25	1264.32	995.77	32	26.97
47	15	50	1488.07	1084.75	163	37.18
48	15	75	1815.22	1252.35	582	44.95
49	15	100	2242.40	1586.20	1265	41.37
50	15	150	2459.52	1691.42	4308	45.41
51	20	25	1660.91	1302.89	35	27.48
52	20	50	1740.66	1304.54	163	33.43
53	20	75	2096.76	1433.07	614	46.31
54	20	100	2226.39	1638.15	1402	35.91
55	20	150	2518.16	1747.42	5879	44.11
56	25	25	1550.71	1311.63	38	18.23
57	25	50	1835.39	1469.78	207	24.88
58	25	75	2276.94	1654.92	821	37.59
59	25	100	2061.86	1575.67	1306	30.86
60	25	150	2347.76	1653.30	4687	42.00
Average			1824.49	1346.20	1260.75	35.53

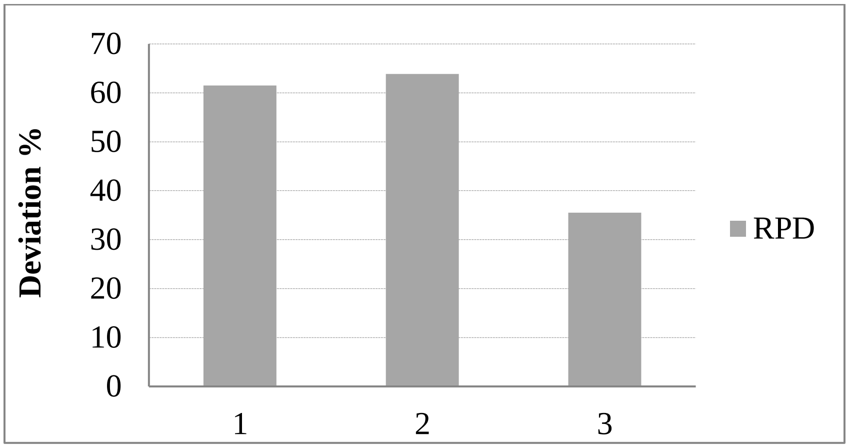

Figure 6.4: The effect of inventory on the heuristic performance

It is obvious from Figure 6.5 that the average total length is decreased when more inventories are available in the network. The increase in the available inventory in the network allows more flexibility in assigning consumers to more different retail stores. This ultimately decreases the total length traveled. It can be noticed that the gap between the MAC and the heuristic increases as I move from relaxed inventory scenario (3) to tighter scenario (1).

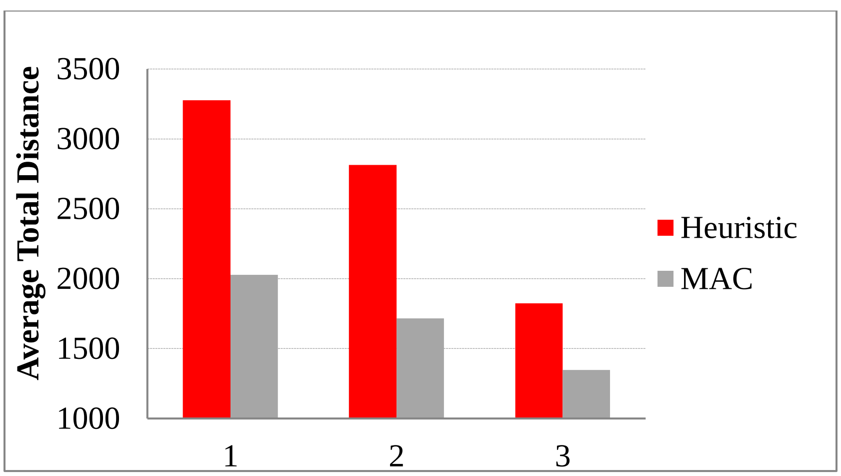

Figure 6.5: The effect of inventory on the average total distance

6.5.2.2 Benefit of Using the Integrated Distribution System

In this section, I illustrate the benefit of using the integrated distribution system instead of the two existing distribution systems. The benefit is calculated by solving the generated problem instances in two settings. In the first setting, the same vehicles are used to simultaneously deliver the products from the depot to the retail stores and from the retail stores to the consumers. However, two different fleets of vehicles are used in the second setting. The first fleet is used to deliver the products from the depot to the retail stores while the second fleet is used to deliver the products from the retail stores to the consumers. In this case, the problem is decomposed into two sub-problems. The first sub-problem is solving the CVRP for distributing the products from the depot to the retail stores. The second sub-problem is solving the PDP for distributing the products from the retail stores to the consumers. The sum of traveling costs for these two sub-problems represents the solution for the second setting.

The results of 60 problem instances under the two settings are calculated using the MAC and the heuristic and presented in Table 6.6. The results are classified according to the three available inventory scenarios. The results reported under the first setting (i.e., the integrated distribution system) are the same as those reported in Tables 6.3–6.5. The results reported under second setting (i.e., the two existing distribution systems) are the combined tour lengths obtained after solving two sub-problems. It can be noticed that the total cost (total length) increases significantly when the two existing distribution systems are used. In this case, a retail store is visited twice, once for delivering its orders and once for collecting the consumers' orders. Visiting a retail store twice increases the total traveling cost. The results reported in Table 6.6 show that the average total tour length increases by 33% and 44% (for the heuristic and MAC, respectively) when the two existing distribution systems are used. The high percentage increase in tour length clearly exhibits the benefit of the proposed integration of the distribution system over the two existing separate distribution systems.

Table 6.6: The benefit of using the proposed distribution system instead of the existing distribution systems

Available inventory scenario	Heuristic average total length			MAC average total length		
	Integrated distribution system	2 different distribution systems	Increase percentage	Integrated distribution system	2 different distribution systems	Increase percentage
1	3275.61	4227.80	29.07	2028.30	2831.42	39.60
2	2812.91	3744.13	33.10	1716.91	2490.31	45.05
3	1824.49	2615.66	43.36	1346.20	2042.87	51.75
Average	2637.67	3529.19	33.80	1697.14	2454.87	44.65

6.6 Conclusion

I introduce the VRP in omni-channel retailing distribution systems, which is a new variant of the VRP. In the proposed problem, the same fleet of vehicles is used to distribute the products ordered online along with the products required by the retail stores. It integrates two existing distribution systems: the retail distribution system and the consumer distribution system. The new problem can be considered a generalization of both the CVRP and PDP. I provide a mathematical formulation to describe the problem and propose two solution approaches. Bench mark problems are generated to test the effectiveness of the proposed solution approaches. The MAC algorithm produced good quality solutions when compared to the optimal solutions found by solving the mathematical model using CPLEX. The MAC produced an average relative percentage deviation of 0.16% with respect to the optimal solution. The MAC algorithm improves the solutions of all problem instances. The numerical experiments indicate that on average, the proposed MAC algorithm produces better results than the proposed heuristic. In addition, it maintains its high performance in harder problems where the available inventory is tight. I illustrate the benefit of using the proposed integrated distribution system instead of the two existing distribution systems. The numerical experiments show that the proposed integrated distribution system reduces the distribution cost by up to 44%. This is the first construction of a VRP structure for omni-channel retail distribution systems. Our work has paved the way for the investigation of more mathematical models to tackle operational decisions in the omni-channel distribution system. The variant of VRP presented in this chapter is a new problem and no previous results are available and I hope that our research provides a bench mark for future research.

Chapter 7

Conclusion

7.1 Summary

The research presented in this thesis is motivated from the supply chain management of omni-channel retailing distribution systems. Different scenarios are investigated to provide applicable solutions for the optimization of omni-channel retailing distribution networks. In omni-channel retailing, consumers can choose to order products to be delivered at home or to visit a store physically to buy the products. It allows consumers to go shopping seven days a week, twenty-four hours a day. The Delivery of the online ordered products to consumers is the responsibility of the company.

I propose different VRP models arising in omni-channel distribution systems. These models provide practical solutions for the supply chain management in retail industry. These models are considered as new variants of the VRP. They share some characteristics with other VRP variants (different VRP variants are reviewed in chapter 2). However, they have other unique features that are designed to fit the constraints imposed in the omni-channel retail distribution systems e.g. serving different types of customers using the same fleet (retailers and consumers), considering assignment decisions based on inventory availability. The new VRP variants have not been tackled in the VRP literature. I provide brief description for each of the new variants and provide mathematical formulations for these new variants of the VRP. I solve the proposed mathematical models to obtain optimum solutions for small problem instances. The VRP and its variants are NP-hard problems and difficult to solve in the case of large problem instances. Therefore, different heuristics and metaheuristics are proposed to obtain near optimum solutions for large problem instances.

The thesis starts with the model of MCVRP in chapter 3. The multi compartment feature is utilized in one of the proposed VRP models for omni-channel retailing distribution. The MCVRP is already presented in literature but the solution methods are not efficient. In this thesis, I provide an efficient AC algorithm to solve the MCVRP. The proposed AC algorithm produces better results using less computational time. In addition, it maintains its high performance in larger problems as well [66].

After designing an efficient algorithm for MCVRP, I introduce a simple VRP model in omni-channel distribution system in chapter 4. This model is concerned with the design of the optimum travelling salesman route to deliver the online ordered products from the retail stores to the customers. Exploring this model provides better understanding of the VRP model in omni-channel retail distribution systems. I present a mathematical formulation for the problem and solve it to optimality for small size problem instances. In addition, I provide a heuristic and a metaheuristic to solve the large problem instances.

The thesis then considers more complex situation in the omni-channel retail distribution systems by studying the whole distribution network in chapter 5. The distribution network of the company includes two distribution systems; retail distribution system and consumer distribution system. These distribution systems perform two types of deliveries; deliveries from the warehouse to the retail stores and deliveries from the retail stores to the consumers. The model assumes that both distribution systems can be served using the same fleet of vehicles. The products required by the consumers will be stored in different compartment of the same vehicle. Thus, handling the products required by the consumers does not interfere with the products required by the retail stores. Therefore, vehicles with multiple compartments are used in this model. The MCVRP model considered in chapter 3 offers a comprehensive understanding on developing solution methods for MCVRP model in omni-channel distribution systems. I describe the problem and provide a mathematical formulation for the problem. I propose different heuristics and a metaheuristic to solve the large problem instances.

Finally, I introduce another VRP model for the omni-channel retail distribution systems in chapter 6. The models considered in chapters 3 and 4 assume that the consumers are already assigned to retail stores. This means that, the retail stores that satisfy the demand of the consumers are decided and known in advance. However, the last model considers the assignment of consumers to retail stores as a decision variable in the routing problem. In this case, the retail stores satisfying the consumer orders will be decided by the inventory availability at the retail stores. This decision can be made simultaneously while selecting the best routes. I introduce a mathematical formulation for the problem and solve the mathematical model to obtain the optimum solution for small problem instances. In addition, I provide a two-phase heuristic and a metaheuristic to obtain near optimum solutions for large problem instances. Moreover, I generate new bench mark problem instances to evaluate the performance of the proposed algorithms [115].

7.2 Future Research Directions

This thesis investigates new VRP models that arise in omni-channel retailing distribution systems. These models have not been considered in the literature and are considered for the first time in this thesis. The objective of introducing the new models is to provide practical solutions for the supply chain management in retail industry.

Extensions of the proposed models will simulate more real-life applications. One of the extensions is to allow the customers to order items from more than retail store in the same order. This adds more complexity to the problem since more precedence constraints are added. In this case, all the retailers that satisfy the consumer order should be visited before the consumers and by the same vehicle which might add more delivery cost. Otherwise, the company may choose to satisfy the consumer demand in more than one delivery which might reduce the customer satisfaction.

Another extension is to consider the preferred delivery time by the customers. This can be done by assigning time windows to the online orders. Tight time windows may result in reduced utilization of the vehicles which means increased delivery cost. Soft time windows may be a good option for the companies in this condition. It will allow ignoring some customers preferred times those will to increase the cost remarkably. However, this will reduce the customer satisfaction as well.

Although these extensions increase the customer satisfaction, it increases the transportation cost at the same time. The companies should consider this tradeoff between the cost and the customer satisfaction. Pareto analysis may be used in this case to provide a group of solutions with different combinations of the delivery cost and the customer satisfaction. This will enable each organization to select its own level of customer satisfaction along with the transportation cost.

Bibliography

[1] P. Toth and D. Vigo, The vehicle routing problem, SIAM monograph on discrete mathematics and applications, Philadelphia (2014) 1-26.

[2] A.S. Russo, and S.P., Shah, Packaging taxes and recycling incentives: the German Green Dot program. National Tax Journal, 47 (1994) 689-701.

[3] G.B. Dantzig, and J.H. Ramser, The truck dispatching problem, Management Science, 6 (1959) 80–91.

[4] Deloitte (2017). Global Powers of Retailing: The art and science of customers. Retrieved from Deloitte website: www.deloitte.com.

[5] P. C. Verhoef, P. K. Kannan, and J. J. Inman, From multi-channel retailing to omni-channel retailing: introduction to the special issue on multi-channel retailing. Journal of Retailing, 91 (2015) 174–181.

[6] S. Lorini, J.Y. Potvin, and N. Zufferey, Online vehicle routing and scheduling with dynamic travel times, Computers & Operations Research 38 (2011) 1086–1090.

[7] C.K.Y. Lin, A vehicle routing problem with pickup and delivery time windows, and coordination of transportable resources, Computers & Operations Research 38 (2011) 1596–1609.

[8] R. Baldacci, E. A. Hadjiconstantinou and A. Mingozzi, An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. Operations Research, 52 (2004) 723–738.

[9] R. Fukasawa, H. Longo, J. Lysgaard, M. P. Aragão, M. Reis, E. Uchoa, and R. F. Werneck, Robust branch-and-cut-and-price for the capacitated vehicle routing problem. Mathematical Programming, 106(3), (2006) 491–511.

[10] A. N. Letchford, and J. J. Salazar-González, Stronger multi-commodity flow formulations of the Capacitated Vehicle Routing Problem. European Journal of Operational Research, 244(3), (2015) 730–738.

[11] U. Derigs, and R. Kaiser, Applying the attribute based hill climber heuristic to the vehicle routing problem. European Journal of Operational Research, 177, (2007) 719–732.

[12] T. J. Ai, and V. Kachitvichyanukul, Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem. Computers & Industrial Engineering, 56, (2009) 380–387.

[13] Y. Nagata, and O. Bräysy, Edge assembly-based memetic algorithm for the capacitated vehicle routing problem. Networks, 54, (2009) 205–215.

[14] W. Y. Szeto, Y. Wu, and S. C. Ho, An artificial bee colony algorithm for the capacitated vehicle routing problem. European Journal of Operational Research, 215, (2011) 126–135.

[15] M. Reimann, K. Doerner, and R.F. Hartl, D-Ants: Savings Based Ants divide and conquer the vehicle routing problem. Computers & Operations Research, 31, (2004) 563–591.

[16] B. Yu, Z. Yanga, and B. Yao, An improved ant colony optimization for vehicle routing problem. European Journal of Operational Research, 196, (2009) 171-176.

[17] N.E. Toklu, L.M. Gambardella, and R. Montemanni, A multiple ant colony system for a vehicle routing problem with time windows and uncertain travel times, Journal of Traffic and Logistics Engineering 2 (2014) 52-58.

[18] J. Brandao, A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem, Computers & Operations Research 38 (2011) 140–151.

[19] J. F. Cordeau, G. Laporte, M. W. P. Savelsbergh, and D. Vigo, Vehicle routing.In: C. Barnhart, G. Laporte (Eds.), Transportation, Handbooks in Operations Research and Management Science (2007) 367–428, Elsevier Science Publishers, Amsterdam.

[20] B. L. Golden, S. Raghavan, and E. A. Wasil, The Vehicle Routing Problem: Latest Advances and New Challenges. Operations Research/Computer Science Interfaces Series (2008), Springer, New York.

[21] G. Laporte, Fifty years of vehicle routing. (2009) Transportation Science, 43, 408–416.

[22] B. Eksioglu, A. V. Vural, and A. Reisman, The vehicle routing problem: A taxonomic review. Computers & Industrial Engineering, 57(4), (2009) 1472–1483.

[23] R. Lahyani, M. Khemakhem, and F. Semet, Rich vehicle routing problems: From a taxonomy to a definition. European Journal of Operational Research, 241 (2015) 1–14.

[24] N. Azi, M. Gendreau, J.Y. Potvin, An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles, European Journal of Operational Research 202 (2010) 756–763.

[25] R. Macedo, C. Alves, J.M.V. Carvalho, F. Clautiaux, S. Hanafi, Solving the vehicle routing problem with time windows and multiple routes exactly using a pseudo-polynomial model, European Journal of Operational Research 214 (2011) 536–545.

[26] D. Cattaruzza, N. Absi, D. Feillet, T. Vidal, A memetic algorithm for the Multi Trip Vehicle Routing Problem, European Journal of Operational Research 236 (2014) 833–848.

[27] M. Reed, A. Yiannakou, and R. Evering, An ant colony algorithm for the multi-compartment vehicle routing problem, Applied Soft Computing 15 (2014) 169–176.

[28] D. Cattaruzza, N. Absi, D. Feillet, D. Vigo, An iterated local search for the multi-commodity multi-trip vehicle routing problem with time windows, Computers & Operations Research 51 (2014) 257–267.

[29] M. Salani, I. Vacca, Branch and price for the vehicle routing problem with discrete split deliveries and time windows, European Journal of Operational Research 213 (2011) 470–477.

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

[31] J.W. Escobar, R. Linfati, P. Toth, M.G. Baldoquin, A hybrid granular tabu search algorithm for the multi-depot vehicle routing problem, J Heuristics DOI 10.1007/s10732-014-9247-0.

[32] C. Contardo, R. Martinelli, A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints, Discrete Optimization 12 (2014) 129–146.

[33] D. Gulczynski, B. Golden, E. Wasil, The multi-depot split delivery vehicle routing problem: An integer programming-based heuristic, new test problems, and computational results, Computers & Industrial Engineering 61 (2011) 794–804.

[34] S. Salhi, A. Imran, N.A. Wassan, The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulation and a variable neighborhood search implementation, Computers & Operations Research 52 (2014) 315–325.

[35] R. G. Dondo, J. Cerdá, A hybrid local improvement algorithm for large-scale multi-depot vehicle routing problems with time windows, Computers and Chemical Engineering 33 (2009) 513–530.

[36] Y. Gajpal, P. Abad, An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup, Computers & Operations Research 36 (2009) 3215 – 3223.

[37] Y. Gajpal, P. Abad, Saving-based algorithms for vehicle routing problem with simultaneous pickup and delivery, Journal of the Operational Research Society 61 (2010) 1498-1509.

[38] A. Subramanian, E. Uchoa, L.S. Ochi, New Lower Bounds for the Vehicle Routing Problem with Simultaneous Pickup and Delivery, Experimental Algorithms, Proceedings of the 9th International Symposium, (2010) 276-87.

[39] E.E. Zachariadis, C.D. Tarantilis, C.T. Kiranoudis, An adaptive memory methodology for the vehicle routing problem with simultaneous pick-ups and deliveries, European Journal of Operational Research 202 (2010) 401–411.

[40] F.P. Goksal, I. Karaoglan, F. Altiparmak, A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery, Computers & Industrial Engineering 65 (2013) 39–53.

[41] H.F. Wang, Y.Y. Chen, A coevolutionary algorithm for the flexible delivery and pickup problem with time windows, Int. J. Production Economics 141 (2013) 4–13.

[42] R. Liu, X. Xie, V. Augusto, C. Rodriguez, Heuristic algorithms for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care, European Journal of Operational Research 230 (2013) 475–486.

[43] F. C. J. Lokin, Procedures for travelling salesman problems with additional constraints. European Journal of Operational Research, 3 (1979) 135–141.

[44] B. Kalantari, A. V. Hill, and S. R. Arora, An algorithm for the traveling salesman problem with pickup and delivery customers. European Journal of Operational Research, 22 (1985) 377–386.

[45] M. W. P. Savelsbergh, and M. Sol, The general pickup and delivery problem. Transportation Science, 29 (1995) 17–29.

[46] Q. Lu, and M. Dessouky, An exact algorithm for the multiple vehicle pickup and delivery problem. Transportation Science, 38 (2004) 503–514.

[47] C. K. Ting, and X. L. Liao, The selective pickup and delivery problem: formulation and a memetic algorithm. International Journal of Production Economics, 141 (2013) 199-211.

[48] S. Ropke, J.F. Cordeau, Branch and cut and price for the pickup and delivery problem with time windows, Transportation Science 43 (2009) 267–286.

[49] R. Baldacci, E. Bartolini, A. Mingozzi, An exact algorithm for the pickup and delivery problem with time windows, Operations Research 59 (2011) 414–426.

[50] W. P. Nanry, and J. W. Barnes, Solving the pickup and delivery problem with time windows using reactive tabu search. Transportation Research, 34 (2000) 107–121.

[51] H. C. Lau, and Z. Liang, Pickup and delivery with time windows: algorithms and test case generation. International Journal of Artificial Intelligence Tools, 11 (2002) 455–472.

[52] H. Li, and A. Lim, A metaheuristic for the pickup and delivery problem with time windows. International Journal of Artificial Intelligence Tools, 12 (2003) 173–186.

[53] G. Pankratz, A grouping genetic algorithm for the pickup and delivery problem with time windows. OR Spectrum, 27 (2005) 21–41.

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

[55] Q. Lu, and M. Dessouky, A new insertion-based construction heuristic for solving the pickup and delivery problem with time windows. European Journal of Operational Research, 175 (2006) 672–687.

[56] C. D’Souza, S. N. Omkar, and J. Senthilnath, Pickup and delivery problem using metaheuristics techniques. Expert Systems with Applications, 39 (2012) 328–334.

[57] R. Masson, S. Ropke, F. Lehuédé, O. Péton, A branch-and-cut-and-price approach for the pickup and delivery problem with shuttle routes, European Journal of Operational Research 236 (2014) 849–862.

[58] A. Rais, F. Alvelos, M.S. Carvalho, New mixed integer-programming model for the pickup-and-delivery problem with transshipment, European Journal of Operational Research 235 (2014) 530–539.

[59] J. Rieck, C. Ehrenberg, J Zimmermann, Many-to-many location-routing with inter-hub transport and multi-commodity pickup-and-delivery, European Journal of Operational Research 236 (2014) 863–878.

[60] G. Berbeglia, J. F. Cordeau, I. Gribkovskaia, and G. Laporte, Static pickup and delivery problems: a classification scheme and survey. Top, 15 (2007) 1–31.

[61] S. N. Parragh, K. F. Doerner, and R. F Hartl, A survey on pickup and delivery problems Part I: Transportation between customers and depot. Journal für Betriebswirtschaft, 58-1 (2008a) 21–51.

[62] S. N. Parragh, K. F. Doerner, and R. F Hartl, A survey on pickup and delivery problems Part II: Transportation between pickup and delivery locations. Journal für Betriebswirtschaft, 58-2 (2008b) 81–117.

[63] M. Dorigo, V. Maniezzo, and A. Colorni, Ant system: Optimization by a colony of cooperating agents, IEEE Transaction son Systems, Man, and Cybernetics—Part B, 26-1 (1996) 1–13.

[64] B. Bullnheimer, R.F. Hartl, and C. Strauss, An improved ant system algorithm for the vehicle routing problem, Annals of Operations Research 89 (1999) 319–328.

[65] Y. Bin, Y.Z. Zhen, and Y. Baozhen, An improved ant colony optimization for vehicle routing problem, European Journal of Operational Research 196 (2009) 171–176.

[66] M.M.S. Abdulkader, Y. Gajpal, and T.Y. ElMekkawy, Hybridized ant colony algorithm for the Multi Compartment Vehicle Routing Problem. Applied Soft Computing 37 (2015) 196–203

[67] R. Montemanni, L.M. Gambardella, A.E. Rizzoli, and A.V. Donati, Ant colony system for a dynamic vehicle routing problem, Journal of Combinatorial Optimization, 10 (2005) 327–343.

[68] S.R. Balseiro, I. Loiseau, and J. Ramonet, An ant colony algorithm hybridized with insertion heuristics for the Time dependent vehicle routing problem with time windows, Computers & Operations Research 38 (2011) 954–966.

[69] J.J. De la Cruz, C.D.P. Arboleda, V. Cantillo, J.R.M. Torres, A two-pheromone trail ant colony system—tabu search approach for the heterogeneous vehicle routing

problem with time windows and multiple products, Journal of Heuristics 19 (2013) 233–252.

[70] J. H. Holland, Adaptation in natural and artificial systems. Ann Arbor, MI: University of Michigan Press (1975).

[71] R. M. Brady, Optimization strategies gleaned from biological evolution. Nature, 317 (1985) 804-806.

[72] J. Berger, and M. Barkaoui, A parallel hybrid genetic algorithm for the vehicle routing problem with time windows. Computers & operations research, 31 (2004) 2037-2053.

[73] W. Ho, G. T. Ho, P. Ji, and H. C. Lau, A hybrid genetic algorithm for the multi-depot vehicle routing problem. Engineering Applications of Artificial Intelligence, 21 (2008) 548-557.

[74] P. K. Nguyen, T. G. Crainic, and M. Toulouse, A hybrid generational genetic algorithm for the periodic vehicle routing problem with time windows. Journal of Heuristics, 20 (2014) 383-416.

[75] J. Kennedy and R. Eberhart, "PSO optimization", Proceedings of IEEE International Conference on Neural Networks, IV (1995) 1941-1948.

[76] S. Geetha, G. Poonthalir, and P. T. Vanathi, Nested particle swarm optimisation for multi-depot vehicle routing problem. International Journal of Operational Research, 16 (2013) 329-348.

[77] B. Yao, B. Yu, P. Hu, J. Gao, and M. Zhang, An improved particle swarm optimization for carton heterogeneous vehicle routing problem with a collection depot. Annals of Operations Research, 242 (2016) 303-320.

[78] Y. Marinakis, G. R. Iordanidou, and M. Marinaki, Particle swarm optimization for the vehicle routing problem with stochastic demands. Applied Soft Computing, 13 (2013) 1693-1704.

[79] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, Optimization by simulated annealing. science, 220 (1983) 671-680.

[80] I. H. Osman, Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. Annals of operations research, 41(1993) 421-451.

[81] M. Mirabi, S. F. Ghomi, and F. Jolai, Efficient stochastic hybrid heuristics for the multi-depot vehicle routing problem. Robotics and Computer-Integrated Manufacturing, 26 (2010) 564-569.

[82] W. C. Chiang, and R. A. Russell, Simulated annealing metaheuristics for the vehicle routing problem with time windows. Annals of Operations Research, 63 (1996) 3-27.

[83] C. Wang, D. Mu, F. Zhao, and J. W. Sutherland, A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup–delivery and time windows. Computers & Industrial Engineering, 83 (2015) 111-122.

[84] F. Glover, Heuristics for integer programming using surrogate constraints. Decision Sciences, 8 (1977) 156-166.

[85] J. A. G. Willard, Vehicle routing using r-optimal tabu search. Master's thesis, The ManagementSchool, ImperialCollege, (1989) London.

[86] P. Badeau, F. Guertin, M. Gendreau, J. Y. Potvin, and E. Taillard, A parallel tabu search heuristic for the vehicle routing problem with time windows. Transportation Research Part C: Emerging Technologies, 5 (1997) 109-122.

[87] E.D. Chajakis, and M. Guignard, Scheduling deliveries in vehicles with multiple compartments, Journal of Global Optimisation 26 (2003) 43–78.

[88] P. Avella, M. Boccia, and A. Sforza, Solving a fuel delivery problem by heuristic and exact approaches, European Journal of Operational Research 152 (2004)170–179.

[89] A. El Fallahi, C. Prins, and R.W. Calvo, A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem, Computers & Operations Research 35 (2008) 1725–1741.

[90] L. Muyldermans, G. Pang, On the benefits of co-collection: experiments with a multi-compartment vehicle routing algorithm, European Journal of Operational Research 206 (2010) 93–103.

[91] S. Lin, Computer solutions for the travelling salesman problem. Bell System Technical Journal 44 (1965) 2245–2269.

[92] N. Christofides, A. Mingozzi, P. Toth, The vehicle routing problem, in: Combinatorial Optimization, Wiley, Chichester, (1979) p. 315–338.

[93] E. Brynjolfsson, Y. J. Hu, and M. S. Rahman, Competing in the age of omnichannel retailing. MIT Sloan Management Review, 54 (2013) 23–29.

[94] G. Dantzig, R. Fulkerson, and S. Johnson, Solution of a large-scale traveling-salesman problem. Journal of the operations research society of America, 2 (1954) 393-410.

[95] J. D. Little, K. G. Murty, D. W. Sweeney, and C. Karel, An algorithm for the traveling salesman problem. Operations research, 11 (1963) 972-989.

[96] H. Crowder, and M. W. Padberg, Solving large-scale symmetric travelling salesman problems to optimality. Management Science, 26 (1980) 495-509.

[97] N. Ulder, E. Aarts, H. J. Bandelt, P. V. Laarhoven, and E. Pesch, Genetic local search algorithms for the traveling salesman problem. Parallel problem solving from nature, (1991) 109-116.

[98] G. Laporte, The traveling salesman problem: An overview of exact and approximate algorithms. European Journal of Operational Research, 59 (1992) 231-247.

[99] P. Larranaga, C. M. Kuijpers, R. H. Murga, I. Inza, and S. Dizdarevic, Genetic algorithms for the travelling salesman problem: A review of representations and operators. Artificial Intelligence Review, 13 (1999) 129-170.

[100] D. L. Applegate, R. E. Bixby, V. Chvatal, and W. J. Cook, The traveling salesman problem: a computational study. Princeton university press, (2011).

[101] K. S. Ruland, and E. Y. Rodin, The pickup and delivery problem: Faces and branch-and-cut algorithm. Computers & mathematics with applications, 33 (1997) 1-13.

[102] J. Renaud, F. F. Boctor, and J. Ouenniche, A heuristic for the pickup and delivery traveling salesman problem. Computers & Operations Research, 27 (2000) 905-916.

[103] J. Renaud, F. F. Boctor, and G. Laporte, Perturbation heuristics for the pickup and delivery traveling salesman problem. Computers & Operations Research, 29 (2002) 1129-1141.

[104] I. Dumitrescu, S. Ropke, J. F. Cordeau, and G. Laporte, The traveling salesman problem with pickup and delivery: polyhedral results and a branch-and-cut algorithm. Mathematical Programming, 121 (2010) 269.

[105] M. I. Hosny, and C. L. Mumford, The single vehicle pickup and delivery problem with time windows: intelligent operators for heuristic and metaheuristic algorithms. Journal of Heuristics, 16 (2010) 417-439.

[106] M. Veenstra, K. J. Roodbergen, I. F. Vis, and L. C. Coelho, The pickup and delivery traveling salesman problem with handling costs. European Journal of Operational Research, 257 (2017) 118-132.

[107] M. Bustillo, and A. F. Geoffrey, Wal-Mart Sees Stores as Online Edge. The Wall Street Journal (2009, December). Retrieved from: www.wsj.com

[108] M. R. Shay, and G. Bright, The Economic Impact of the US Retail Industry (2014). Retrieved from: National Retail Federation website: nrf.com/retailsimpact.

[109] D. Herhausen, J. Binder, M. Schoegel, and A. Herrmann, Integrating bricks with clicks: retailer-level and channel-level outcomes of online–offline channel integration. Journal of Retailing, 91 (2015) 309-325.

[110] Truck Regulations (November 2, 2016). Retrieved from: http://www.winnipeg.ca/publicworks/trafficControl/truckRegulations.stm.

[111] R. Baldacci, P. Toth, and D. Vigo, Exact algorithms for routing problems under vehicle capacity constraints. Annals of Operations Research, 175 (2010) 213–245.

[112] S. Mazzeo, and I. Loiseau, An ant colony algorithm for the capacitated vehicle routing. Electronic Notes in Discrete Mathematics, 18 (2004) 181-186.

[113] J. Rowell, Omni-channel retailing. Romanian Distribution Committee Magazine, 4 (2013) 12–15.

[114] G. Clarke, and J. Wright, Scheduling of vehicles from a central depot to a number of delivery points. Operations Research, 12 (1964) 568–581.

[115] M.M.S. Abdulkader, Y. Gajpal, and T.Y. ElMekkawy, Vehicle Routing Problem in Omni-Channel Retailing Distribution Systems. International Journal of Production Economics, Under Review.