EVENT BASED DYNAMIC MULTI-FACILITY VEHICLE ROUTING MODEL

A Thesis by

Ajaykrishna Madhu

Bachelor of Technology in Mechanical Engineering, Amrita Vishwa Vidyapeetham, 2016

Submitted to the Department of Industrial, Systems, and Manufacturing Engineering
and the faculty of the Graduate School of
Wichita State University
in partial fulfillment of
the requirements for the degree of
Master of Science

December 2019

EVENT BASED DYNAMIC MULTI-FACILITY VEHICLE ROUTING MODEL

The following faculty members have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirement for the degree of Master of Science with a major in Industrial Engineering.

Krishna Krishnan, Committee Chair

Saideep Nannapaneni, Committee Member

Rajeev Nair, Committee Member

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor Dr. Krishna Krishnan for the continuous support of my study and related research. His knowledge and experience steered me in the right the direction whenever he thought I needed it. A special thanks to the committee members, Dr. Saideep Nannapaneni and Dr. Rajeev Nair, for their valuable comments and time to be a part of this research. Finally, I would like to thank my family and friends for being patient and supportive through-out my masters. It has been a long and fruitful journey which has motivated me to carry further with the same motivation.

ABSTRACT

The term ‘Industry 4.0’ mark the evolution of Information and Communication Technologies that promoted technological changes enabling modern supply chain logistics systems to work efficiently in a dynamic environment. Industry 4.0 emphasises on interconnected systems through the application of the latest technologies such as the Internet of Things (IoT), Cyber-Physical Systems (CPS), and in this research, these systems are assumed to be deployed which provides a channel for real-time information transfer between the dispatcher and vehicles allowing the logistic system the possibility of reacting to dynamic events such as new service requests and dropouts. This paper focuses on a variant of VRP, the capacitated multi-depot vehicle routing problem which takes into consideration the dynamic nature of the system. Since the VRP is NP-hard, a new 2 stage algorithm is introduced that make use of different combinations of various heuristics and local search metaheuristics to generate high-quality solutions that may not be optimal in a very short time. The proposed methodology creates an initial route plan for the predetermined requests and then modifies the original plan as new events are dynamically revealed over time. Results are collected, and the performance of different algorithms are compared in a dynamic setting.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

Chapter Page

LIST OF TABLES

LIST OF TABLES (continued)

LIST OF FIGURES

LIST OF FIGURES (continued)

CHAPTER 1

INTRODUCTION

The onset of the 21st century saw the evolution and rapid growth of Cyber-Physical Systems (CPS), the Internet of Things (IoT), cloud computing, Information and Communication Technologies (ICT) and its various application in the supply chain systems (SCS). Technological innovations and increased competition have led industries to focus on building an efficient and flexible SCS to cope up with new challenges, which is rapidly changing the industry. This revolution called as the fourth industrial revolution (Industry 4.0) is facilitating the integration of various processes and systems across different sectors which opens up a platform for better communication and coordination in a new smart way, revolutionising SCS to be more efficient and cost-effective (Barreto et al., 2017).

Industry 4.0 is a term that represents the gradual integration of information technology and communication technology into traditional industrial and manufacturing practices. The introduction of computers into the industry which marked the third industrial revolution had a disruptive impact since a new technology was introduced into the conventional industrial space. Now with Industry 4.0, these computers are connected and have the ability to communicate and work together to make smart decisions without human intervention. CPS, IOT, ICT, and other smart technologies have made Industry 4.0 possible and this, in turn, makes the smart factory a reality. Data is the one crucial factor as they make smart machines more intelligent and this results in smart factories become more productive and efficient with very fewer resources getting wasted. The network of digitally connected machines that create and share data makes the backbone of Industry 4.0. The high volume of data collected by these interconnected machines may contain patterns or insights that can be analysed by humans in a reasonable time that can help with performance analysis, proactive measures for maintenance and other issues. This helps to focus immediate attention to issues that are the most detrimental to the system and take corrective steps at the right time to optimise their operations. For example, a connected supply chain can adjust to new information such as a shipment delay caused by unusual weather where the connected system will pro-actively adjust its manufacturing priorities. These changes have promoted the integration of Industry 4.0 in most sectors of SCS, especially in logistics to achieve a modern and agile logistics system.

The customer request for individualised services is increasing day by day contributing to the increased demand uncertainty and therefore to stay relevant in the competitive environment; logistics have an important role to play in a business, which has to account for this demand uncertainty and variability. Therefore, finding the cheapest routes to reduce the transportation cost while catering the dynamic customer service requests help enterprises gain a competitive edge (Cassettari et al., 2018). This is especially true for smaller businesses as larger businesses tend to have lower logistic costs as they could benefit from economies of scale. Moreover, with the accelerated globalisation of commerce, products are being transported over long distances, and that leads to more fossil-fuels being consumed since long-distance movements require an extraordinary amount of energy. Due to the increased push for eco-friendly supply chain systems, transport-related emissions and their adverse effects have attracted many global concerns. According to the 2015 study conducted by the United States Environmental Protection Agency, the major pollutants that contribute to poor air quality are nitrogen oxides (NO_x), volatile organic compounds (VOCs) and particulate matter (PM). The transportation sector contributes to over 50% of total NO_x emissions, 30% of VOCs emissions and 20% of PM emission. Among various freight transportation methods, large trucks are the fastest growing contributor to emissions. So it can be inferred that finding the best and efficient routes can reduce the emissions and thereby make the logistics model an eco-friendly one. The need for optimal routes leads to the Vehicle Routing Problem (VRP) introduced by Danting and Ramser in 1959.

VRP play an essential role in Supply chain logistics management and are very crucial for enterprises in satisfying their customer requests while keeping the overall logistics cost down. VRP can be described as a problem of assigning optimal collection/delivery routes for vehicles from one or more depots to several customer locations subject to various constraints (Jeon et al., 2007). VRP is a combinatorial optimisation problem that is NP-compete and has been studied widely during the past decade. There are several variants of the VRP namely; Capacitated VRP (CVRP), VRP with time windows (VRPTW), VRP with Pickup and Delivery (VRPPD), multi-depot VRP (MDVRP), dynamic VRP (D-VRP) and their combinations (Xu et al., 2018). The adoption of Industry 4.0 and the continued effort by the enterprises to improve service quality and minimise cost have led the dynamic variant of the vehicle routing problem to gain more attention.

In VRP research history, many of the work that has been done assumes static conditions and known information such as customer locations, service requests and vehicle information. However, in reality, due to the recent technological advancements in areas such as communication technologies, Transport management systems (TMS), Intelligent Transportation Systems (ITS), Warehouse Management Systems (WMS) and IoT devices, real-time information exchange have made many of the VRP dynamic; meaning a part of the customer locations and requests will gradually change over time. A well connected and established TMS system can enable data exchange between order management software and depots. These systems once properly implemented can aid businesses to manage their fleet costs, integrate new technologies and handle communications with customers, business partners and other carriers. Also, these systems enable the logistics division to use real-time data such as customer requests and vehicle information to improve the effectiveness and efficiency of the logistic process.TMS system also plays a vital role in incorporating GPS technology to accurately locate its vehicle fleet, consolidate shipments and interact with other smart systems deployed. As more physical entities are equipped with RFID tags, sensors and bar codes, vehicle dispatching centres could monitor the movement of every product from the origin till the destination for the whole supply chain system improving the transparency of the supply chain. With the advancement of technology, the functionalities of the TMS system have significantly improved over the years in a way that it now offers cloud-based services that have enabled better end-to-end supply chain visibility and thereby improving the decision making quality of the management. These improvements have made businesses to serve all their customers better at a low cost while maintaining a high customer satisfaction level.

Another unique field that interoperates in various sectors of transportation is the Intelligent Transportation System (ITS). It operates in various different fields such as transportation infrastructure, policies, logistic control and logistic operations. ITS plays a vital role in ensuring a smooth operation by making use of GPS systems, data processing algorithms, IT, advanced computing techniques, virtual planning and operations. The recent advancement in technology and IT systems have incorporated mobile devices, smart automobiles, and other contextual mobility solutions into ITS which made the system more sustainable, effective, efficient and economical. When combined with Real-time information systems that collect data by various sensors, GPS devices and IoT devices from different nodes in the logistics network, ITS could potentially improve the real-time decision-making ability giving rise to a flexible and efficient system capable of inter-machine communications and cooperative technologies. Considering the logistics scenario under Industry 4.0, a fully fledged ITS can have endless possibilities such as smart delivery management, automated parking, multimodal cargo planning, estimation of CO_2 footprint, fleet driver support, priority and speed synchronisation resulting in an eco-friendly and economic logistic system. Accident, risk-free and high-speed transportation, reliable and efficient logistics operations are some of the areas where ITS projects its strengths. ITS is not limited to road transport but also caters other modes of transport such as water, air and rail.

Another notable technological application is the Warehouse Management Systems (WMS) that plays a vital role in the flow of goods across the supply chain. Warehouses act as a hub through which products are distributed in a supply chain. In the context of Industry 4.0, they also have a vital role to play in the smooth integration of smart technologies into logistics. The inbound and outbound logistics, as well as the warehouse functioning, will see a remarkable transformation with the implementation of WMS into traditional warehouses. With proper adoption and implementation, WMS introduces the concept of smart management that will promise full control, coordination and alignment of all value chain links. The real-time information transfer can help predict shipment arrival, optimise just-in-time delivery, and finally prepare the docking slots for the smooth operation of the logistic system. At the same time, the sensors such as RFID will tell the details of the shipment that is delivered to all the links in the supply chain. WMS also automates the allocation of storage spaces based on the shipment and makes sure the right kind of equipment is available to move these shipments autonomously. The end result is better visibility of on-hand inventory levels which helps the management to improve decision-making such that stockout scenarios can be prevented which might help increase the customer satisfaction levels. These smart systems that improve the vehicle fleet effectiveness and sustainability of the logistic system are forcing businesses that strive to stay relevant in the competitive environment, to implement these systems in the day to day VRP context. The Fig. 1.1 below summarises logistics in the context of Industry 4.0.

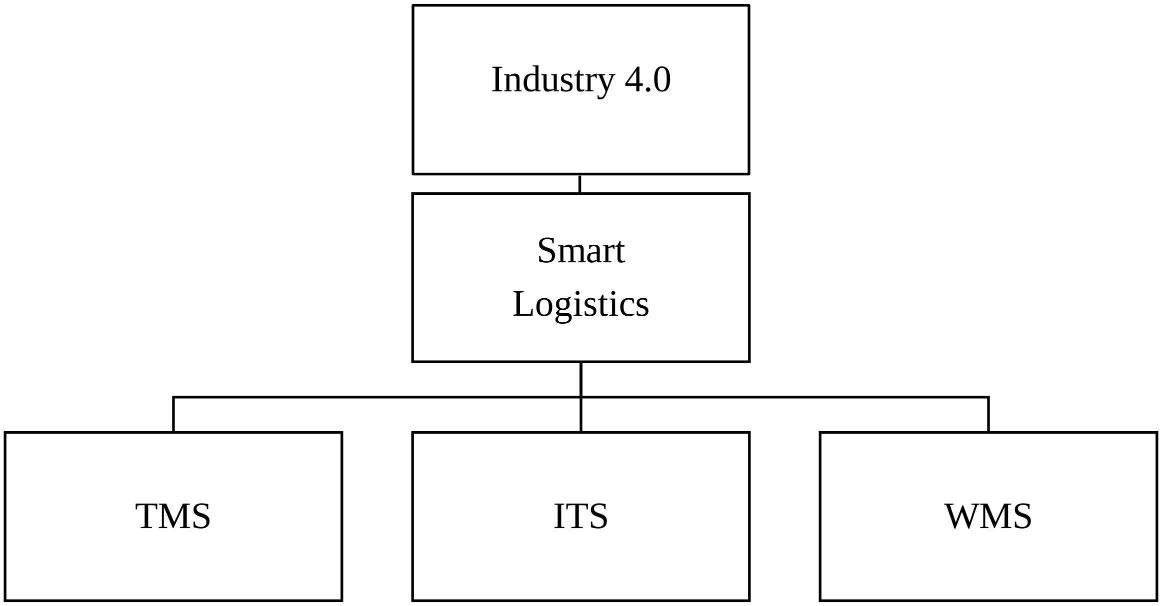

Figure 1.1. Smart logistics components

The number of research focusing on the multi-depot version of the D-VRP is comparatively few as most of the studies in the literature deal with more simpler single depot D-VRP. So, this paper studies the dynamic multi-depot vehicle routing problem (D-MDVRP) for a courier collection service. Each new customer service request or cancellation of an existing customer request is considered an event. Since the D-MDVRP is an NP-hard problem (Toth et al., 2002), heuristics or metaheuristics are better suited as they provide good quality solutions to real-world problems within a reasonable computational time. So, this research aims to develop a 2-stage algorithm using different heuristics, local search strategies and compare the results, computational time to select the best combination and make quick, informed routing decisions to minimise the total route distance while handling static and dynamic events.

The remainder of this paper is organised as follows. Chapter 2 reviews the relevant literature regarding Industry 4.0, VRP, DVRP and D-MDVRP. Chapter 3 explains the methodology, mathematical formulation and solution approach for the D-MDVRP. In Chapter 4, a case study with different scenarios where the proposed methodology is implemented is detailed with their results. Finally, Chapter 6 contains a discussion of the results and ends with a conclusion and future work.

CHAPTER 2

LITERATURE REVIEW

2.1 Industry 4.0

The fourth industrial revolution also called as Industry 4.0 is a concept which encloses the digital transformation of industries through the introduction of Information and Communication technology, digital technology by the combined application of Big Data, IoT, Artificial Intelligence (AI), Cyber-Physical-Systems (CPSs), Cloud Computing, Information Management Systems (IMS) and Advanced Robotics (Barreto et al., 2017). The term 'Industry 4.0' was coined in Germany, but similar digital transformations can be seen in other parts of the world with different labels such as Smart Industry, Smart Factories, Industrial Internet of Things, and Advanced Manufacturing (Tjahjono et al., 2017).

The first industrial revolution is marked by the development of steam engines and mechanical equipment in the late 18^{th} century. The beginning of the 20^{th} century saw the introduction of assembly lines and conveyor belts which marked the second industrial revolution. The third industrial revolution was characterised by the use of computers and IT for electronic automation during the late 20^{th} century. Finally, the integration of cyber technologies into the traditional industrial value chains led to the fourth industrial revolution called Industry 4.0 which is entirely different from the previous three Industrial Revolutions. This era is considered crucial and strategic for major industrial powers, and thus many governments have started supporting the development and applications of these solutions. The benefits include increased profits, lower costs, improved customer satisfaction, reduced delivery times, innovative products, more flexible and efficient processes.

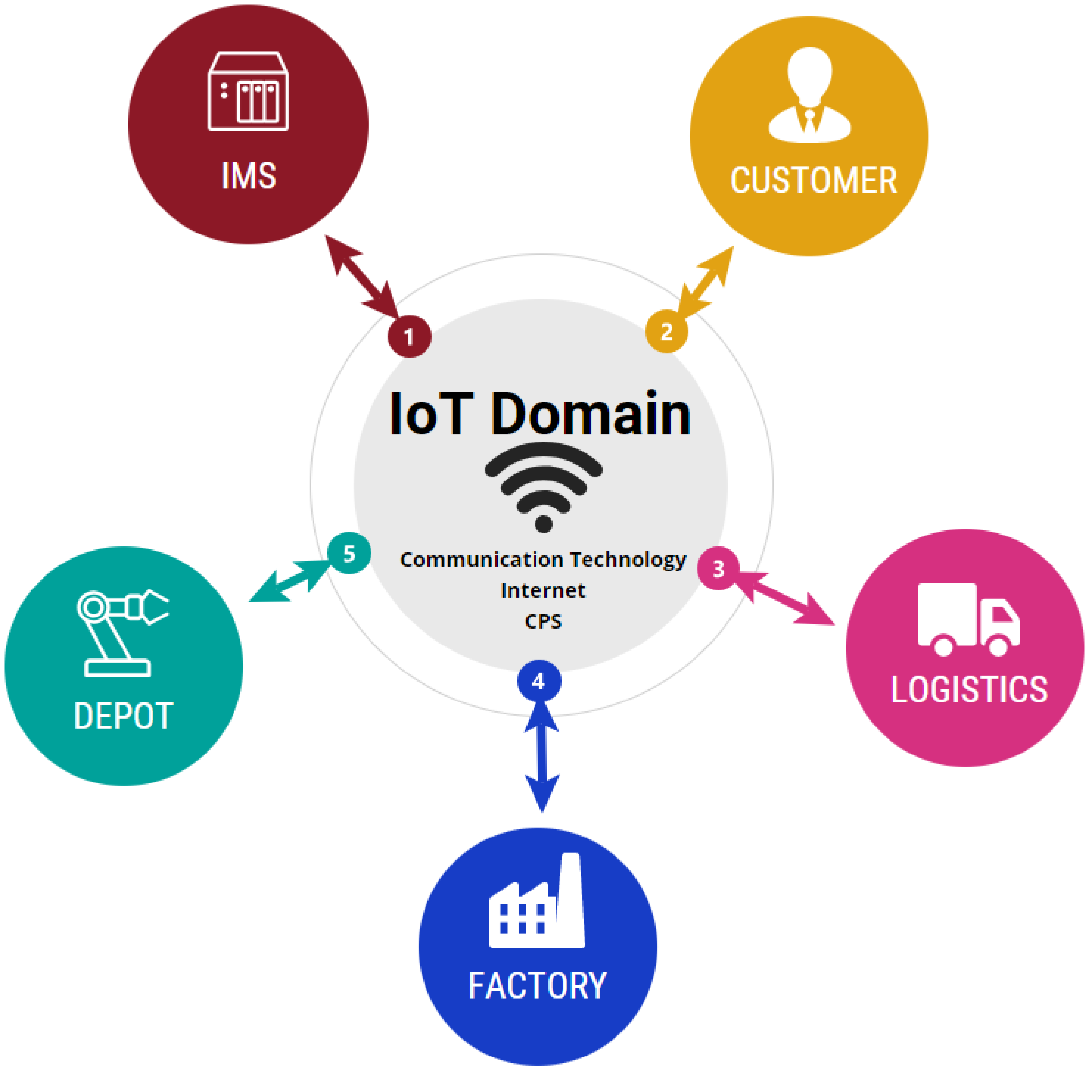

Figure 2.1. Industry 4.0 Framework

Industry 4.0 promotes the use of some of the innovative technologies and systems such as IoT, CPS, AI, Big Data, and Cloud Computing in industries. This use provides a new platform for better communication and cooperation of processes across various sectors effectively enabling smarter ways to automate processes with minimal human intervention. These smart technologies and solutions have also changed the way how the data is collected and analysed, and one of the compelling use cases of Industry 4.0 is the Industrial Digital Twin.

Digital Twin is a virtual entity or a digital profile that represents a real physical system but simultaneously stands on its own to help optimise business performance. The digital copy acts as a twin by collecting and exchanging real-time information from the physical system using IoT devices that can provide crucial information about the system performance throughout its life cycle. This real-time connection between the real and digital world helps to realistically model measurements of variability and uncertainty which is otherwise difficult. Also, making use of cheap, powerful computing architecture that is available, this massive amount of data can be processed using advanced algorithms to obtain real-time predictive feedback which can enable fundamental design and process variations to optimise the current system. These factors and favourable data storage costs have expanded the possibilities to introduce digital twin into businesses which in turn drives business value.

2.2 Dynamic Vehicle Routing Problem

The vehicle routing problem (VRP) is a combinatorial optimisation problem that originated from the Travelling Salesman Problem (TSP) which has been studied for a long time. Dantzig et al., (1959) proposed the first VRP to model the delivery of gasoline. VRP aims to find the set of optimal routes between depots and geographically dispersed customer locations to transport goods with the minimum logistic costs. Each of the customer locations is to be served by exactly one vehicle from the fleet, and the total demand should not exceed the total fleet capacity. The distances between customer locations, depots are considered as Euclidean distance between the coordinate pair. Finding an optimal solution for a VRP is NP-Hard, and so heuristics and metaheuristics are preferred over exact solutions to solve real-world problems. Several studies have been done on variants of VRP using different solution methods. Laporte et al., (1985) proposed the first exact algorithm for MDVRP which is a branch and bound algorithm. Then in 1988 they modified the branch and bound algorithm to solve asymmetric MDVRP (Laporte et al., 1988). In the subsequent year's several approximate solution methods were developed. Bertsimas et al., (1991) analysed various methods to minimise service wait times for different traffic situations. Renaud et al., (1996) proposed a Tabu Search algorithm to solve an MDVRP with capacity and route length restrictions. Kilby et al., (1999) applied the Guided Local Search meta-heuristic which shares similarity with Tabu search to VRP. Wu et al., (2002) used a Simulated Annealing (SA) heuristics to minimise the cost for an MDVRP. Giosa et al., (2002) developed a “cluster first, route second” strategy for the MDVRP with Time Windows (MDVRPTW), and compared the computational time for various clustering techniques. Nagy et al., (2005) proposed several enhancements to an integrated heuristic method for solving the MDVRP with pickup and deliveries. All the works mentioned above are static VRPs. However, in reality, most of the practical VRPs are dynamic.

In recent years, due to the advancement of ICT, real-time fleet management has become a possibility. This advancement combined with robust, cheap computing architecture and advanced algorithms aids in developing real-time solutions for dynamic situations (Ghiani et al., 2003). With the recent integration of Industry 4.0, real-time information exchange between the vehicles and the depots is seamless and cost-effective. Whenever a new customer request arrives, the back-end software gathers real-time fleet information, analyses the data, and gives a quick action plan to the dispatchers who sent the proper course of action to the fleet drivers. Also, GPS systems and ICT are used to get the current location of vehicles, and this allows the dispatchers to monitor and manage the fleet effectively. These systems also allow the dispatchers to know the real-time distance between various vehicles and visualise them to make sure the fleet is fully utilised for better reaction time. Fig. 2.2 illustrates the communication network for the Dynamic-MDVRP case studied in this research.

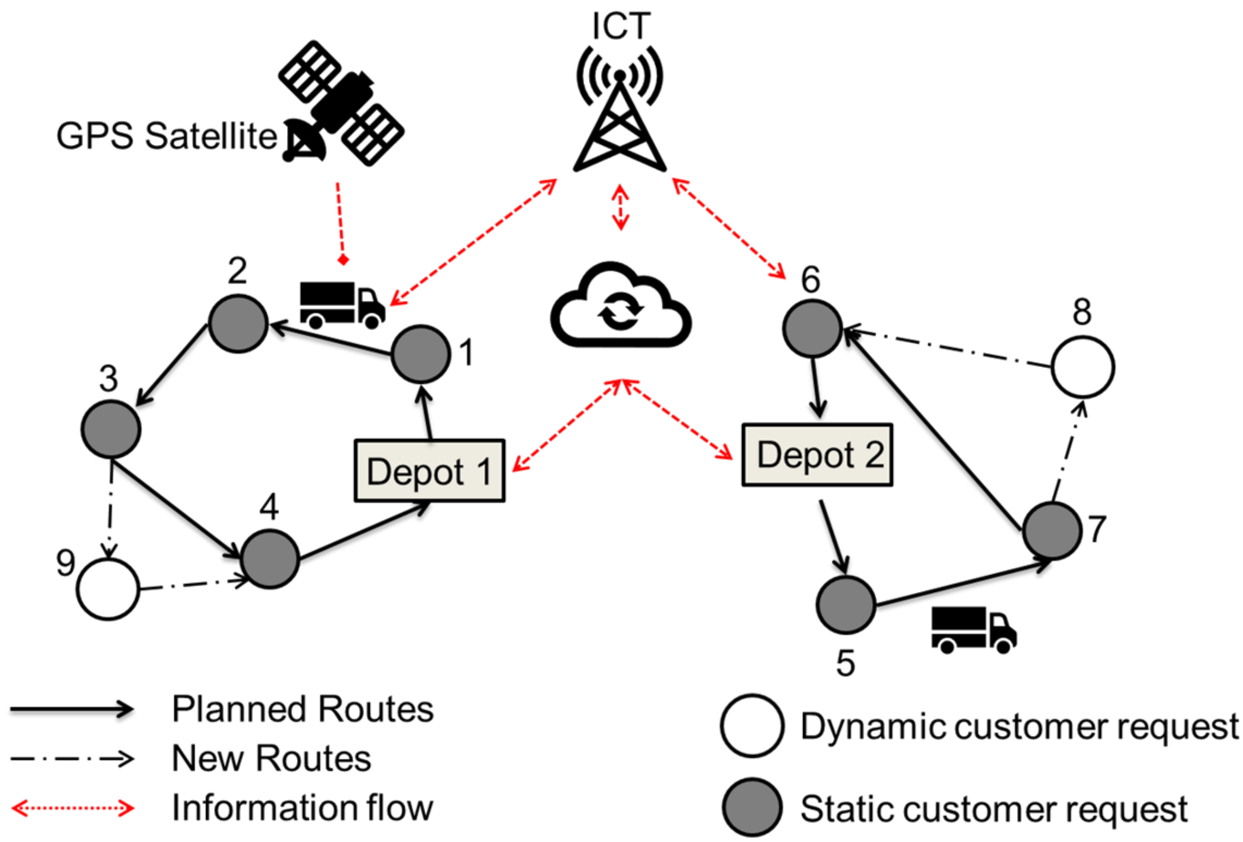

Figure 2.2. A communication network for Dynamic-MDVRP

There is numerous research that focuses on the Dynamic VRPs which take into account the variability and uncertainty in data. Here are some of the DVRPs from the past decade found in the literature. Mitrović-Minić et al., (2004) analysed a dynamic pick up and delivery problem using a double-horizon based heuristics with long term and short term goals. A Dial-a-Ride problem was studied by Attanasio et al., (2004), who used a parallel Tabu Search algorithm. Haghani et al., (2005) proposed a genetic algorithm for a DVRP with time-dependent travel times. One of the notable contributions in this research area is by Okhrin et al., (2008). This paper is a DVRP problem with time windows, and it incorporates the possibility to react to dynamic events through the use of information and communication system. They modelled the problem as a MILP and developed a genetic algorithm to solve it.

2.3 Related work

D-MDVRP is a combination of D-VRP and MDVRP where the dynamic customer requests are satisfied with vehicles from multiple depots. Barceló et al., (2007) integrated the VRP model and the dynamic traffic simulation to develop a routing-simulation modelling framework to incorporate dynamism from traffic conditions. Other works combined solution strategies such as the work done by Timmermann et al., (2008), who used an adaptive neighbourhood search combined with a Tabu search heuristic to solve an MDVRPTW. However, it assumes that a vehicle only revises their routes after serving the next customer. A dynamic pick-up and delivery problem were investigated by Kuo et al.,(2014) in 2014, who developed an insertion heuristic to solve a D-MDVRP with minimal changes in the original route plan when dynamic customer requests arrive. The research aimed to compare the results between when current vehicle positions are considered and when they are not and concluded that considering real-time locations reduced the completion time but increased the rejection rate. Around the same time, AbdElAziz et al., (2014) developed a heuristic method based Cluster-First Route-Second methodology for solving a problem model that handles static and dynamic demands from customers using a homogenous vehicle fleet. Another heuristics was proposed by Meesuptaweekoon et al., (2014) in 2014 that had 2 phases comprising of route construction and vehicle dispatch. The dispatch time of each vehicle from the homogenous fleet is determined by maximising the waiting time to provide the opportunity to add more arriving customers in the future. Recently, Yang et al., (2017) extended the ant colony based heuristics to dynamic vehicle routing problems with time windows. Moreover, in 2018, Xu et al., (2018) proposed a hybrid ant colony based clustering approach to deal with the dynamic problem of MDVRP quickly.

CHAPTER 3

METHODOLOGY

This chapter defines the problem and explains the various assumptions and methodology used to find the solution. The latter part of this chapter explains the mathematical formulation and briefly describe the various algorithms used to solve the problem.

In a supply chain network for a courier collection service, there will be pickup requests originating from geographically dispersed customer locations and these requests should be satisfied from different depots with the help of a vehicle fleet. In order to stay competitive, the business needs to serve all the customer requests while keeping the logistics costs low. Lowering the logistics cost include finding the optimal routes each time there is a new customer request or a request cancellation even while the vehicles are already in transit. This study focuses on Dynamic-MDVRP model incorporating real-time vehicle locations, vehicle capacities and pickup requests to manage a heterogeneous vehicle fleet for a courier collection business. The research aims to reduce the total distance travelled by the business’s vehicle fleet while catering to all the customer's service requests. At the beginning of the planning period, some customer service requests and their locations are available. The dispatcher creates an initial route plan for the known customer requests based on the available data. Over time, new requests and their locations get revealed, or some of the existing customer requests gets cancelled both of which are termed as an event while the vehicles are already in transit. The original route plan is modified as each new event arrives to incorporate changes to the existing route. All the customer requests are accepted as it assumed that the sum of customer requests is always less than the total fleet capacity. The request from a single customer is assumed to be satisfied by a single vehicle from the fleet. Every vehicle is assigned to a specific depot, and all their routes must start and end at their respective depots. It is also assumed that there is a constant exchange of information between the dispatcher and the vehicles with the help of ICT. The following section looks into a mathematical formulation to solve an MDVRP.

3.1 Problem formulation

The following MDVRP is formulated as an integer programming model based on the generic MDVRP formulation presented by Golden et al.,(1974).

Minimise

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{N} d_{ij} x_{ij}^{k} \tag{4. 1}$$

Subject to

$$\sum_{i=1}^{n} \sum_{k=1}^{N} x_{ij}^{k} = 1 \quad (\mathrm{j} = \mathrm{M} + 1, \ldots\ldots, \mathrm{n}) \tag{4. 2}$$

$$\sum_{j=1}^{n} \sum_{k=1}^{N} x_{ij}^{k} = 1 \quad (\mathrm{i} = \mathrm{M} + 1, \ldots\ldots, \mathrm{n}) \tag{4. 3}$$

$$\sum_{i=1}^{n} x_{ip}^{k} - \sum_{j=1}^{n} x_{pj}^{k} = 0 \quad (\mathrm{k} = 1, \ldots\ldots, \mathrm{N}) \;\; (p = 1, \ldots\ldots, N) \tag{4. 4}$$

$$\sum_{i=1}^{n} Q_i \left(\sum_{j=1}^{n} x_{ij}^{k} \right) \le p_k \quad (k = 1, \ldots\ldots, N) \tag{4. 5}$$

$$\sum_{i=1}^{M} \sum_{j=M+1}^{n} x_{ij}^{k} \le 1 \quad (\mathrm{k} = 1, \ldots\ldots, \mathrm{N}) \tag{4. 6}$$

$$\sum_{p=1}^{M} \sum_{i=M+1}^{n} x_{ip}^{k} \le 1 \quad (\mathrm{k} = 1, \ldots\ldots, \mathrm{N}) \tag{4. 7}$$

$$x_{ij}^{k} = 0 \; or \; 1 \; for \; all \; i, j, k \tag{4. 8}$$

$$x \in S \tag{4. 9}$$

Where

$$x_{ij}^{k} = \begin{cases} 1 \; if \; arc \; (i, j) \; is \; traversed \; by \; vehicle \; k \\ \qquad\qquad 0 \; otherwise \end{cases}$$

$n \quad = number \; of \; nodes \quad (nodes \; 1, \ldots\ldots, M \; are \; depots)$

$N \quad = number\ of\ vehicles$

$P_k \ = capacity\ of\ vehicle\ k$

$Q_i \ = demand\ at\ node\ i$

$d_{ij} = shortest\ distance\ from\ node\ i\ to\ node\ j$

The objective function is to minimise the total distance travelled by all vehicles given by Equation (4.1). Equations (4.2) and (4.3) ensures that each node is visited by only one vehicle. Route continuity is represented by Equation (4.4). Equation (4.5) represents vehicle capacity constraints. Equations (4.6) and (4.7) restricts the number of vehicles to available vehicles.

The above mentioned MDVRP formulation gives optimal routes for static VRP cases where customer locations and pickup quantities are known before planning the routes. To handle the real-time customer requests whose pickup requests arrives after the route planning, the initial routes are modified using the MDVRP formulation with the new data as the input to obtain optimal routes. Essentially, each time a new event (i.e. new pickup request or cancellation of existing request) occurs, a static MDVRP formulation is solved to get the optimal route for that specific scenario. As mentioned in the first section, finding the optimal solution for an MDVRP is NP-Hard which limits the size of problems that can be solved using exact algorithms since the search space for all the feasible routes will be large, and this will cause long computational times which might not be practical. When it comes to finding the solution for a Dynamic-MDVRP, this issue is further amplified. In the case of D-MDVRP, as each new event arrives, a fast and reasonable solution to incorporate the changes to the existing routes is expected as the vehicles are already in transit. Delays in updating routes can cause the business to lose money, customer satisfaction level requests may drop and eventually, they might even lose the competitive edge. Therefore, the necessity for quick, good quality solutions leads to heuristics and metaheuristics which are the solution approaches that are preferable over the exact solution methods, since they provide near-optimal solutions in a very short time. The solution approach proposed to handle predetermined, and real-time customer service in this study includes a 2-stage algorithm developed in Python software that uses a combination of different heuristics and local search strategies to provide a fast and high-quality solution. So, an initial route plan is developed using the 2-stage algorithm for all the static customers whose locations and pickup quantities are known before the route is planned. When new events start to appear, the 2-stage algorithm is again used to find an updated solution using the new data for each event. The next section describes the 2-stage algorithm in detail.

3.2 Solution Approach

As mentioned in the first section, this research aims to develop a 2-stage algorithm using different heuristics, local search strategies and compare the results, computational time to select the best combination and make quick, informed routing decisions to minimise the total route distance while handling static and dynamic events.

In the 2-stage algorithm, the first stage gives the user the option to choose from 3 different heuristics namely; Path Cheapest Arc algorithm, Clarke & Wright Savings algorithm and Christofides algorithm. Based on the algorithm the user defines, the first stage will provide a quick initial solution or routes for all the vehicles. This sub-optimal solution is then fed into the second stage of the algorithm which gives the user the option to choose from 3 different local search metaheuristics which can improve on the current solution namely; Guided Local Search, Simulated Annealing and Tabu Search. The user can also control the algorithm run time in the second stage. So, depending on the algorithm and runtime the user selects, the second stage might improve the first solution to provide an even better result. Each time a new event arrives, the current location coordinates of the vehicles and their capacities are updated and is fed into the 2-stage algorithm which provides a new solution. The following Fig. 3.1 illustrates how the algorithm works.

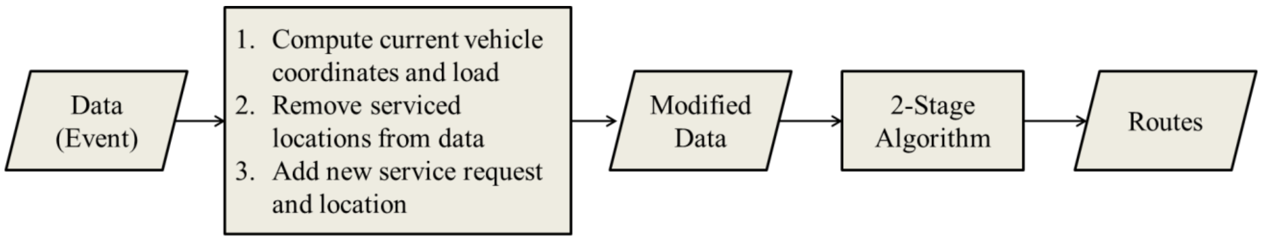

Figure 3.1. Algorithm flowchart

3.3 2-Stage Algorithm

The 2-stage algorithm as mentioned in the previous system has the option to choose between 3 different first solution heuristics namely; Path Cheapest Arc algorithm, Clarke & Wright Savings algorithm and Christofides algorithm.

Path Cheapest Algorithm: This algorithm is more of a rule of thumb where, beginning from the route starting node, the algorithm connects to the next node which produces the shortest route segment, then extends the route by iterating on the last node added to the route.

Clarke & Wright Savings algorithm: Clark and Wright published a heuristic algorithm in 1964 to solve the classical vehicle routing problem (Clarke and Wright, 1964). The algorithm provides a relatively quick and good solution that is not optimal. The algorithm works as follows; the first step calculates the savings for all pairs of customers and these customer points are sorted in the descending order of savings called the savings list. In the next step, the algorithm starts to process each pair of points from the top of the list. For the pair i-j considered, the routes that connect i and j are combined without removing a previously established direct connection between two customer nodes, and without the demand exceeding the vehicle capacity. The below Fig. 3.2 illustrates the savings concept where D is the start/end node of the route.

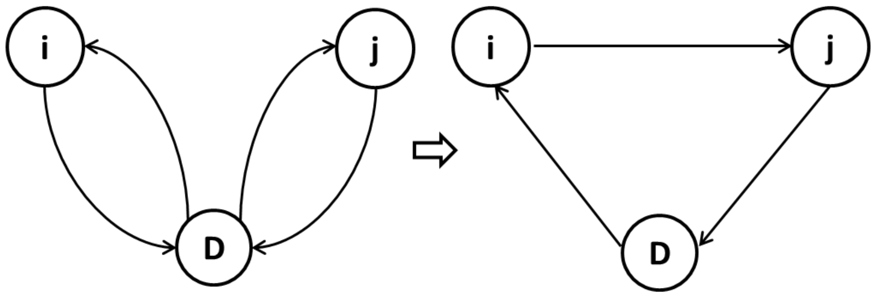

Figure 3.2. Illustration of Savings concept

Christofides algorithm: The Christofides algorithm was developed by N Christofides in 1976 to find a near optimal solution for the Travelling Salesman Problem (Christofides, 1976). The algorithm consists of the following four steps; the first step finds the minimum spanning tree T. The second steps identify the subset of edges M without common vertices from a connected graph that has precisely one connection to all vertices from the vertices with odd degree. The third step combines the edges of M and T to make a multigraph G. Final step develops a Euler cycle in G by skipping the vertices already seen.

The results obtained from the first stage is fed into the second stage of the algorithm that has the option to choose between 3 different local search heuristics namely; Guided Local Search, Simulated Annealing and Tabu Search.

Guided Local Search: it makes use of a penalised cost function to escape local minima and plateaus (Voudouris, 1998). Iterative calls are made to the local search, and whenever the algorithm gets caught at a local minimum, it modifies the cost function by changing the penalties, and the local search is called again to minimise the modified cost function. Since the penalty terms influence the Local search, the search focuses its attention to a better solution in the search space.

Simulated Annealing: SA makes use of decreasing temperatures to balance exploration and exploitation during the search (Metropolis et al., 1953). The algorithm works by generating random 2-opt moves, where if the move improves the current solution, the move is accepted. For the moves that do not improve the current solution, they are accepted with a probability $e^{\frac{-\Delta}{T}}$, where Δ is the change in cost due to the move and T is the current temperature.

Tabu Search: TA makes use of a restriction technique to escape local minima (Glover, 1986). The algorithm begins with the initial solution and with each iteration, it makes local moves to improve the current solution. The new solution is the best solution in the neighbourhood. Each time a new solution is found the old movements are saved to a tabu list so that recently visited solution will not be revisited. This list acts as a restriction mechanism to avoid local optima.

CHAPTER 4

CASE STUDY

In this chapter, two case studies are developed to validate and assess the performance of the 2-stage solution algorithm proposed in the previous chapter.

4.1 Case study 1

The case study developed mimics a supply chain network for a courier collection agency, and the proposed methodology is implemented to find the best result for the VRP. The developed network consists of two depots and several customer locations scattered in the Euclidian plane as seen in Fig. 4.1. In this case study, there are two vehicles for each depot which are used to collect the products from different customer locations, and there is a total of 27 customer locations where 22 customer locations are considered predetermined, and the five customer locations are revealed over time. Real-time information in the form of customer service requests, customer locations, instantaneous vehicle capacities, instantaneous vehicle locations and time are used as inputs for the case study.

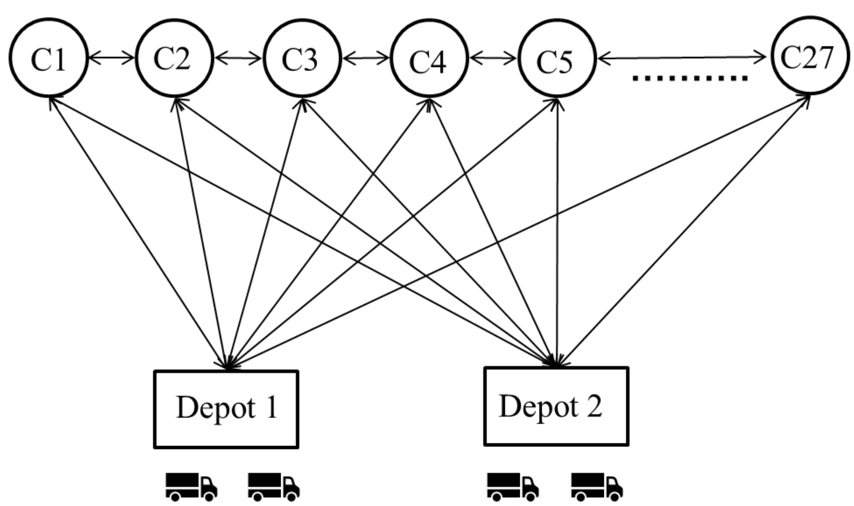

Figure 4.1. The network developed for the case study

The proposed 2-stage algorithm that is implemented in the case study has been coded in Python v3.7 using the routing libraries from OR-Tools, which is an open-source software suite that provides a platform for solving more general routing problems that contain constraints beyond those of a pure TSP. This case study simulates a single period, and the depots are assumed to cater all the customer service requests using any of the four vehicles. Table 1 shows the customer details.

Table 4.1. Customer details

Customer	**Pickup Quantities**	**Location Coordinates**	**Time**
1	3	(26,32)	0
2	5	(6,43)	0
3	2	(21,49)	0
4	3	(20,16)	0
5	3	(15,21)	0
6	5	(41,18)	0
7	2	(39,41)	0
8	3	(14,33)	0
9	5	(25,21)	0
10	4	(30,16)	0
11	3	(37,33)	0
12	4	(22,41)	0
13	1	(30,2)	0
14	5	(8,48)	0
15	4	(15,6)	0
16	4	(15,48)	0
17	5	(19,37)	0
18	3	(15,43)	0
19	5	(42,33)	0
20	3	(38,2)	0
21	3	(35,50)	0
22	2	(10,30)	0
23	3	(5,5)	13
24	4	(5,38)	31
25	1	(19,45)	45
26	6	(45,28)	51
27	4	(35,35)	66

The above table contains the customer pickup quantities, their location and the time when the pickup requests arrived. Time 0 denotes the known customer requests that came in before the routes were planned and all the requests that have time other than 0 are real-time customer requests. Each depot has two vehicles associated with it, which have different capacities. The below table shows the vehicle location coordinates and capacities.

Table 4.2. Vehicle and Depot details

Depot	Vehicle	Location Coordinates	Vehicle Capacity
1	1	(30,40)	30
1	2	(30,40)	20
2	3	(20,25)	30
2	4	(20,25)	20

Table 2 shows the depot locations, which is also the start/end locations for the vehicles. The vehicle capacity column shows that a heterogenous fleet of vehicles is used for the case study. All these data are stored as Excel files and read by the Python code. The excel files mimic the real-time information that the dispatcher uses in a real-world scenario. In the case study, two scenarios are being studied. The first scenario is analyzing the first stage of the 2-stage algorithm where only the algorithms in the first stage are run and the results tabulated. Then for the second scenario, algorithms from both the stages in the 2-stage algorithm are run, and results tabulated.

4.1.1 Scenario – 1

In this section, the data from the tables shown above are used as input to obtain the results by running only the first stage of the 2-stage algorithm. The second stage of the algorithm is not considered in this scenario. There are three heuristics available in the first stage, namely, Path Cheapest Arc algorithm, Clarke & Wright Savings algorithm, Christofides algorithm. These first-stage algorithms produce routes within a fraction of a second and these act as the initial solution for the second stage algorithms. For this scenario, these three first stage algorithms are individually run and then the results in terms of distance (in miles), are tabulated in Table 3.

Table 4.3. First stage results

Algorithm	**Vehicle 1**	**Vehicle 2**	**Vehicle 3**	**Vehicle 4**	**Total Distance**
PATH_CHEAPEST_ARC	75.3	145	61	99	381
SAVINGS (Clarke & Wright)	112	164	87.6	98.3	462.9
CHRISTOFIDES	85.8	84.3	136.3	61.6	368

Table 3 shows the distance traveled by each vehicle and the total distance traveled for each algorithm. Comparing the results obtained, it can be seen that Christofides algorithm produces the best result where the total distance traveled by the vehicle fleet is 368 miles and the worst result being produced by the Savings algorithm where the total distance traveled by the vehicle fleet is 462.9 miles. Path Cheapest Arc algorithm produced a result close to the best one where the total distance traveled by the vehicle fleet is 381 miles. Fig. 4.2 gives a pictorial representation of the distance traveled by each vehicle when using different algorithms.

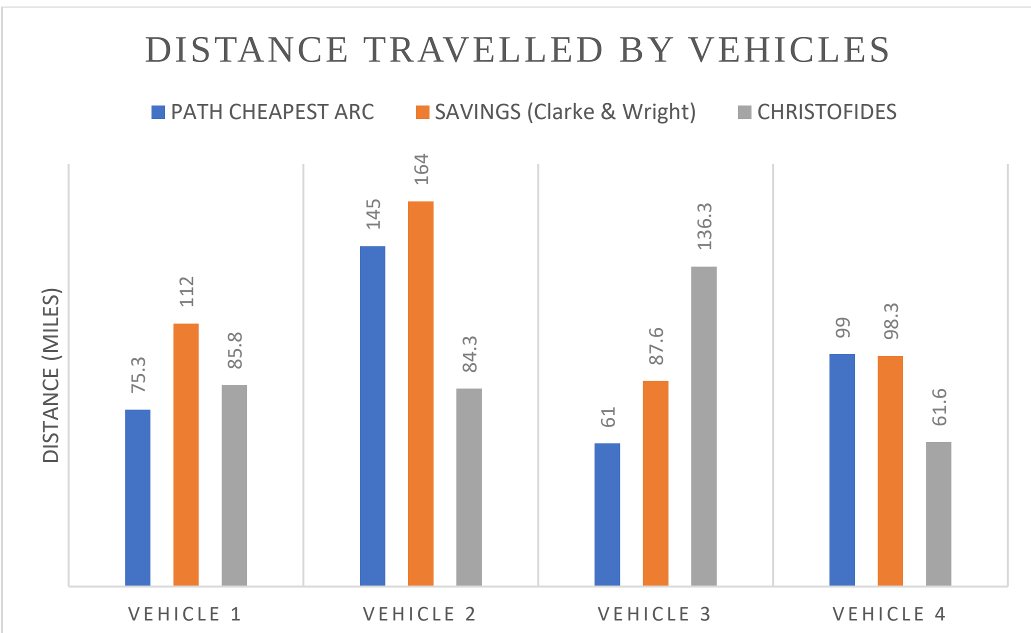

Figure 4.2. Distance comparison chart

4.1.2 Scenario – 2

In this section, the data from the tables shown above are used as input to obtain the results by running both the first stage and the second stage of the 2-stage algorithm. There are three heuristics available in the first stage namely; Path Cheapest Arc algorithm, Clarke & Wright Savings algorithm, Christofides algorithm which are combined with the three local search metaheuristics from the second stage namely; Guided Local Search, Simulated Annealing and Tabu Search. The results produced by the first stage acts as the initial solution for the second stage algorithms. For this scenario, various combinations of the algorithms from the first and second stage are run and then the results in terms of distance (in miles), are tabulated for each combination. The three first stage algorithms and three second stage algorithms produce nine different combinations for the 2-stage algorithm. Each combination of algorithms is run for different lengths of time namely; 0.5 seconds, 1 second, 2 seconds, 5 seconds and their results tabulated.

4.1.2.1 Combination 1 - Path Cheapest Arc and Simulated Annealing

In the first combination, the Path cheapest arc is used in the first stage, and the results from this stage are fed into the second stage that uses Simulated Annealing to improve on the current solution. There results for the different algorithm run times are tabulated in Table 4.

Table 4.4. Combination 1 results

Algorithm Run time (Seconds)	Vehicle 1	Vehicle 2	Vehicle 3	Vehicle 4	Total Distance
0.5	125.2	122.2	61.7	84.7	393.8
1	125.2	122.2	61.7	84.7	393.8
2	125.2	122.2	61.7	84.7	393.8
5	125.2	122.2	61.7	84.7	393.8

The first combination does not show any variation in the distance travelled by vehicle with the change in algorithm runtime. The best result produced is 393.8 miles.

4.1.2.2 Combination 2 - Path Cheapest Arc and Tabu Search

In the second combination, the Path Cheapest Arc algorithm is used in the first stage, and the results from this stage are fed into the second stage that uses Tabu Search to improve on the current solution. There results for the different algorithm run times are tabulated in Table 5.

Table 4.5 Combination 2 results

Algorithm Run time (Seconds)	Vehicle 1	Vehicle 2	Vehicle 3	Vehicle 4	Total Distance
0.5	64.6	79.6	133.2	88.2	365.6
1	82.4	99.2	87.5	75.9	345
2	103.6	78.7	91.7	94.8	368.8
5	103.6	78.7	91.7	94.8	368.8

The second stage of the algorithm shows variation in the results produced with a change in algorithm run time. Fig. 4.3 provides a pictorial representation of the variation of the total distance for the various algorithm run time. The result seems to improve and give the best result of 345 miles when the run time changes from 0.5 seconds to 1 second. However, the results worsen when moving to 2 seconds run time, and there is no change between 2 second and 5 seconds run time.

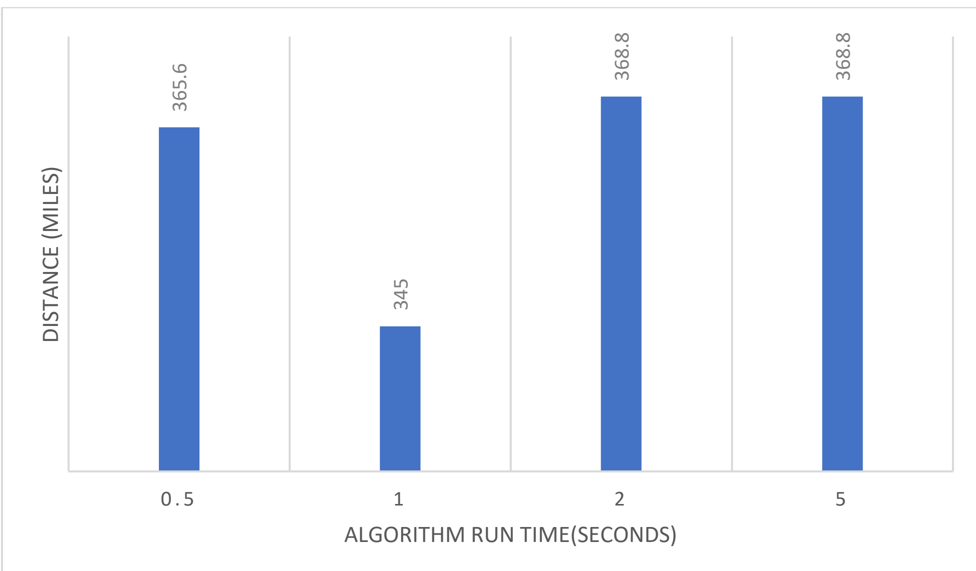

Figure 4.3. Total distance - Combination 2

4.1.2.3 Combination 3 - Path Cheapest Arc and Guided Local Search

In the third combination, the Path cheapest arc is used in the first stage, and the results from this stage are fed into the second stage that uses Guided Local Search to improve on the current solution. There results for the different algorithm run times are tabulated in Table 6.

Table 4.6. Combination 3 results

Algorithm Run time (Seconds)	**Vehicle 1**	**Vehicle 2**	**Vehicle 3**	**Vehicle 4**	**Total Distance**
0.5	83	78.1	45.9	102.2	309.2
1	65.1	83	102.2	45.9	296.2
2	65.1	83	102.2	45.9	296.2
5	65.1	83	102.2	45.9	296.2

The Guided Local Search algorithm shows variation in the results produced with a change in algorithm run time. Fig. 4.4 shows a graphical representation of the variation of the total distance for the various algorithm run time. The result seems to improve and gives the best result of 296.2 miles when the run time changes from 0.5 seconds to 1 second. However, the result remains the same for 2 seconds and 5 second run time.

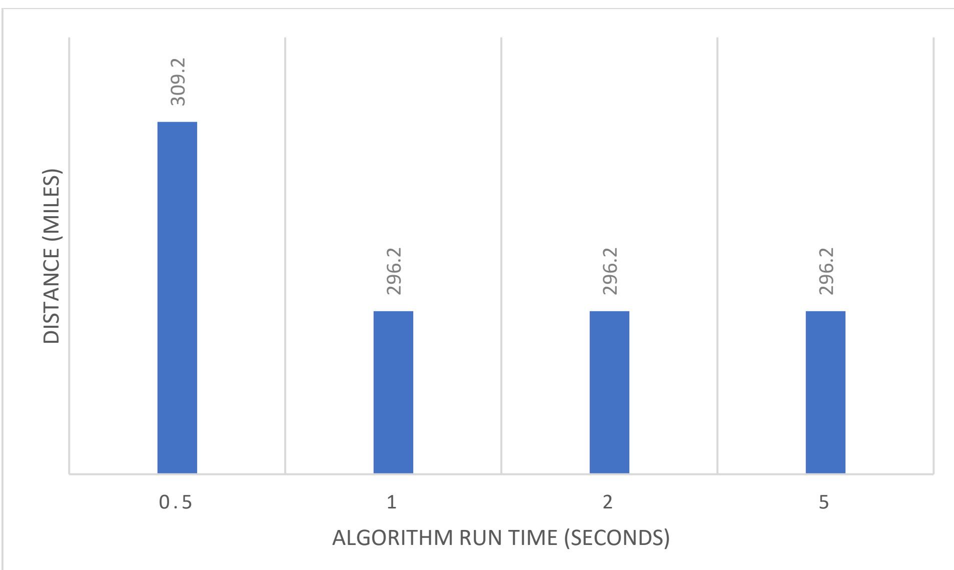

Figure 4.4. Total distance - Combination 3

4.1.2.4 Combination 4 – Savings and Simulated Annealing

In the fourth combination, the Clarke & Wright Savings Algorithm is used in the first stage, and the results from this stage are fed into the second stage that uses Simulated Annealing to improve on the current solution. The results for the different algorithm run times are tabulated in Table 7

Table 4.7. Combination 4 results

Algorithm Run time (Seconds)	**Vehicle 1**	**Vehicle 2**	**Vehicle 3**	**Vehicle 4**	**Total Distance**
0.5	83.7	109.5	84	115.3	392.5
1	83.7	109.5	84	115.3	392.5
2	83.7	109.5	84	115.3	392.5
5	83.7	109.5	84	115.3	392.5

This combination does not show any variation in the distance travelled by vehicle with the change in algorithm runtime. The best result produced is 392.5 miles.

4.1.2.5 Combination 5 – Savings and Tabu Search

In the fifth combination, the Clarke & Wright Savings Algorithm is used in the first stage, and the results from this stage are fed into the second stage that uses Tabu Search to improve on the current solution. The results for the different algorithm run times are tabulated in Table 8.

Table 4.8. Combination 5 results

Algorithm Run time (Seconds)	**Vehicle 1**	**Vehicle 2**	**Vehicle 3**	**Vehicle 4**	**Total Distance**
0.5	98.5	127.3	81	77	383.8
1	98.5	127.3	81	77	383.8
2	98.5	127.3	81	77	383.8
5	80.3	143.2	37	91.9	352.4

The Tabu Search algorithm shows variation in the results produced with a change in algorithm run time. Fig. 4.5 shows a graphical representation of the variation of the total distance for the various algorithm run time. The result seems to stay the same for 0.5, 1 and 2 second run time and then improve to obtain the best result of 352.4 miles for the 5 seconds run time.

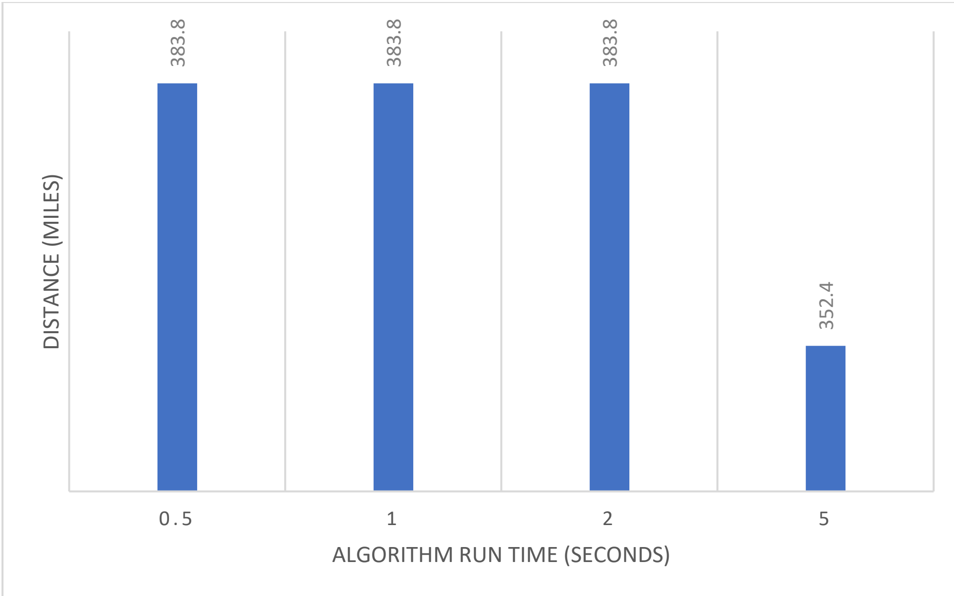

Figure 4.5. Total distance - Combination 5

4.1.2.6 Combination 6 – Savings and Guided Local Search

In the sixth combination, the Clarke & Wright Savings Algorithm is used in the first stage, and the results from this stage are fed into the second stage that uses Guided Local Search to improve on the current solution. The results for the different algorithm run times are tabulated in Table 9.

Table 4.9. Combination 6 results

Algorithm Run time (Seconds)	Vehicle 1	Vehicle 2	Vehicle 3	Vehicle 4	Total Distance
0.5	82.8	68.7	85.5	127	364
1	82.8	68.7	85.5	127	364
2	82.8	68.7	85.5	127	364
5	85.7	90.5	94.8	81.8	352.8

The Guided Local Search algorithm shows variation in the results produced with a change in algorithm run time. Fig. 4.6 shows a graphical representation of the variation. The result seems to stay the same for 0.5, 1 and 2 second run time and then improve to obtain the best result of 352.8 miles for the 5 seconds run time.

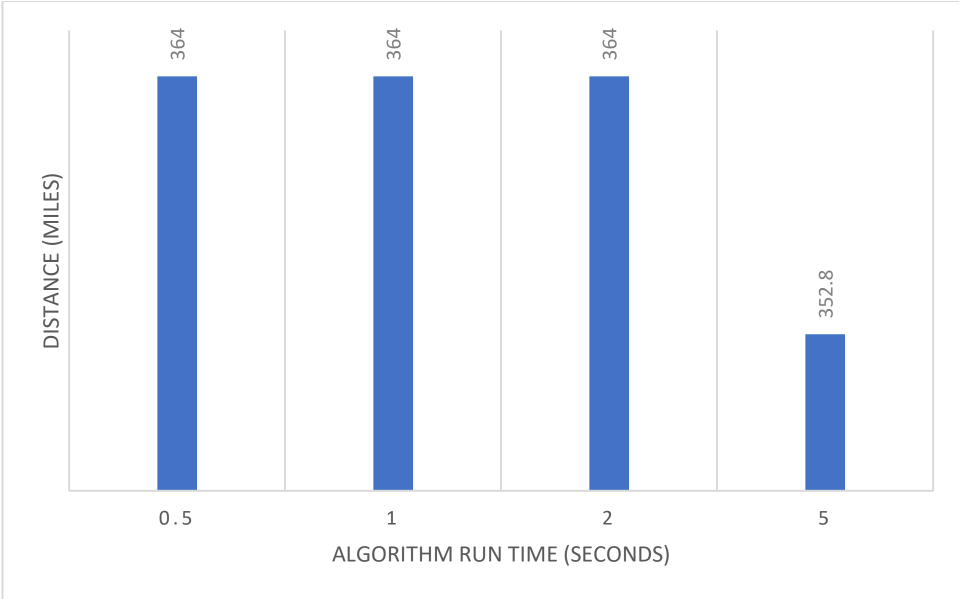

Figure 4.6. Total distance - Combination 6

4.1.2.7 Combination 7 – Christofides and Simulated Annealing

In the seventh combination, the Christofides Algorithm is used in the first stage, and the results from this stage are fed into the second stage that uses Simulated Annealing to improve on the current solution. There results for the different algorithm run times are tabulated in Table 10.

Table 4.10. Combination 7 results

Algorithm Run time (Seconds)	**Vehicle 1**	**Vehicle 2**	**Vehicle 3**	**Vehicle 4**	**Total Distance**
0.5	68.7	75.8	67.4	128	339.9
1	68.7	75.8	67.4	128	339.9
2	68.7	75.8	67.4	128	339.9
5	68.7	75.8	67.4	128	339.9

This combination does not show any variation in the distance travelled by vehicle with the change in algorithm runtime. The best result produced is 339.9 miles.

4.1.2.8 Combination 8 – Christofides and Tabu Search

In the eighth combination, the Christofides Algorithm is used in the first stage, and the results from this stage are fed into the second stage that uses Tabu Search to improve on the current solution. There results for the different algorithm run times are tabulated in Table 11.

Table 4.11. Combination 8 results

Algorithm Run time (Seconds)	Vehicle 1	Vehicle 2	Vehicle 3	Vehicle 4	Total Distance
0.5	81.7	57.4	95.3	120.3	364.7
1	60.7	93.1	89.2	102.2	345.2
2	60.7	93.1	89.2	102.2	345.2
5	60.7	93.1	89.2	102.2	345.2

The second stage of the algorithm shows variation in the results produced with a change in algorithm run time. Fig. 4.11 provides a pictorial representation of the variation of the total distance for the various algorithm run time. The result seems to improve and give the best result of 345.2 miles when the run time changes from 0.5 seconds to 1 second. However, there is no change in results for 2 and 5 second run time.

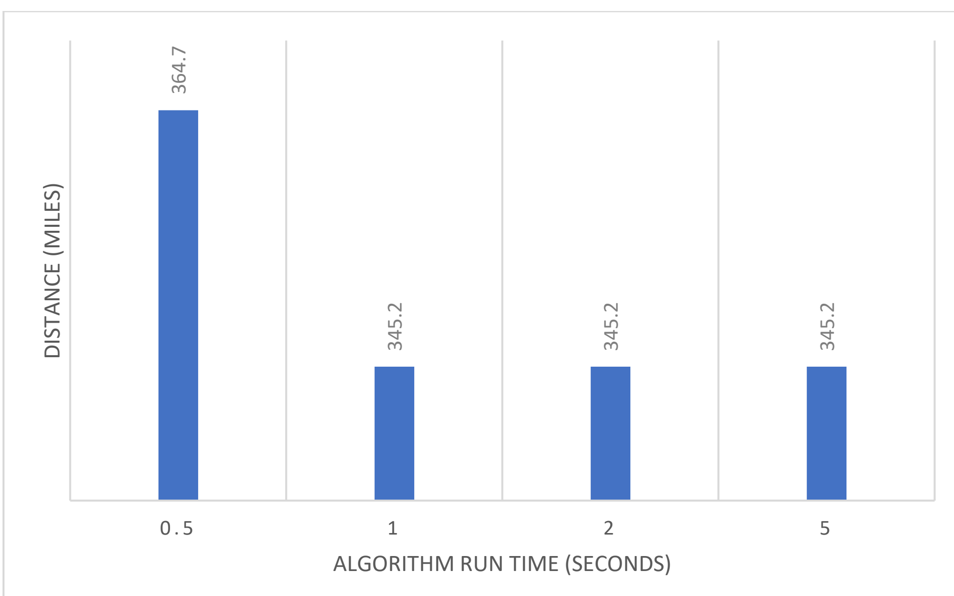

Figure 4.11. Total distance - Combination 8

4.1.2.9 Combination 9 – Christofides and Guided Local Search

In the ninth and final combination, the Christofides Algorithm is used in the first stage, and the results from this stage are fed into the second stage that uses Guided Local Search to improve on the current solution. There results for the different algorithm run times are tabulated below.

Table 4.12. Combination 9 results

Algorithm Run time (Seconds)	Vehicle 1	Vehicle 2	Vehicle 3	Vehicle 4	Total Distance
0.5	86.6	100.3	102.2	55.2	344.3
1	83	105.1	102.2	55.2	345.5
2	77.3	62.3	129.9	90.7	360.2
5	76	103.3	64.4	112.2	355.9

The second stage of the algorithm shows variation in the results produced with a change in algorithm run time, as seen in Fig. 4.12. The result seems to worsen when the run time changes from 0.5 seconds to 1 second. The same trend continues as the result again worsens when moving to 2 second runtime. However, the result seems to improve when the run time is 5 seconds. The best result is 344.3 miles obtained for 0.5 second runtime.

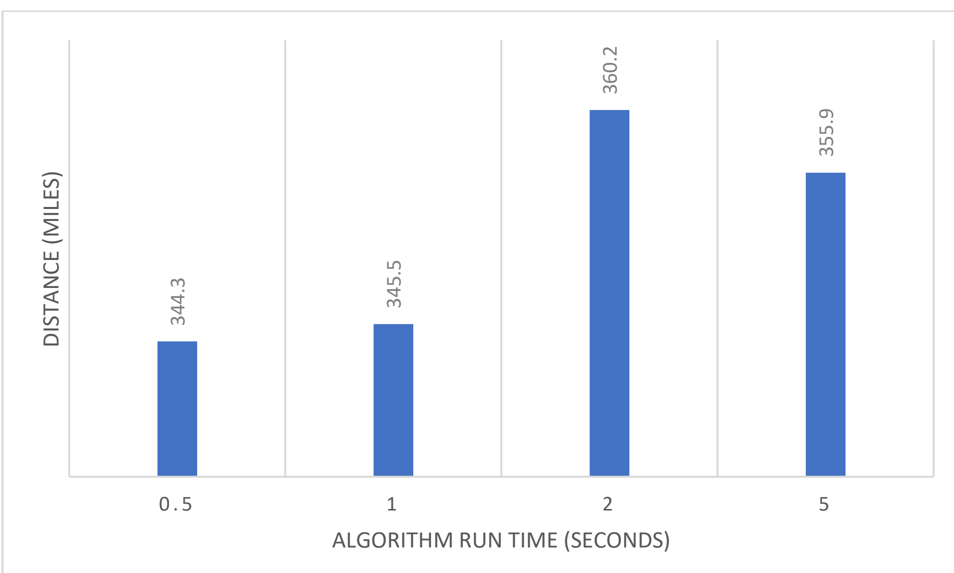

Figure 4.12. Total distance - Combination 9

4.2 Case Study 2

The previous case study developed assessed the performance of the 2-stage decision algorithm on a dataset with 27 customers, two depots and four vehicles. Among the total of 27 customer locations, 22 customer locations are considered predetermined, and the five customer locations are revealed over time. Real-time information in the form of customer service requests, customer locations, instantaneous vehicle capacities, instantaneous vehicle locations, and time are used as inputs for the case study. The second case study is developed to mimic a supply chain network for a courier collection agency with large number of customers that includes both predetermined and dynamic customers. Compared to the first case study, this case study has a total of 148 customers among which 60 customers are predetermined, and 88 customers are revealed over time. The demand increased from 95 units to 909 units/ All the real-time information systems in the form of customer service requests, customer locations, instantaneous vehicle capacities, instantaneous vehicle locations, and time are assumed to be implemented in this case study. The vehicle capacities, number of vehicles per depot and depot location have been carried over from the first case study. Contrary to the first case study, the predetermined demands and the dynamic demands are spread over a span of 7 hours. The below table shows the type and number of customers for each hour.

Table 4.13. Number of customer arrivals

Hour	Predetermined Customers	Dynamic customers
0	22	7
1	8	15
2	4	10
3	9	12
4	7	15
5	6	11
6	4	18

Similar to the first case study, the proposed 2-stage algorithm has been coded in Python v3.7 using the routing libraries from OR-Tools, which is an open-source software suite that provides a platform for solving more general routing problems that contain constraints beyond those of a pure TSP. The depot locations and start/end location for the vehicle remain the same as that in case study 1 and is shown in Table. The same heterogenous fleet of vehicles is used for this case study. All these data are stored as Excel files and read by the Python code. The excel files mimic the real-time information that the dispatcher uses in a real-world scenario.

Similar to the previous case study, the data from the excel files are used as input to obtain the results by running both the first stage and the second stage of the 2-stage algorithm. There are three heuristics available in the first stage, namely, Path Cheapest Arc algorithm, Clarke & Wright Savings algorithm, Christofides algorithm, which are combined with the three local search metaheuristics from the second stage, namely; Guided Local Search, Simulated Annealing and Tabu Search. The results produced by the first stage acts as the initial solution for the second stage algorithms. For this scenario, various combinations of the algorithms from the first and second stages are run and then the results in terms of distance (in miles), are tabulated for each combination. The three first stage algorithms and three second stage algorithms produce nine different combinations for the 2-stage algorithm. Each combination of algorithms is run for different lengths of time, namely; 0.5 seconds, 1 second, 2 seconds, 5 seconds and their results tabulated. This case study does not take into account the performance of individual first stage algorithms as they were already discussed in the previous case study. However, the second case study takes into account the unsatisfied demand and number of dropped customers along with the distance traveled by the vehicle fleet. The results of the decision algorithm are tabulated in tables 14 and 15, and the final results and cost analysis are discussed in the next chapter.

Fig. 4.13 gives a pictorial representation of the 148 customer locations and 2 depot locations on a cartesian plane. The blue dots are the customer locations and red boxes are the depot locations.

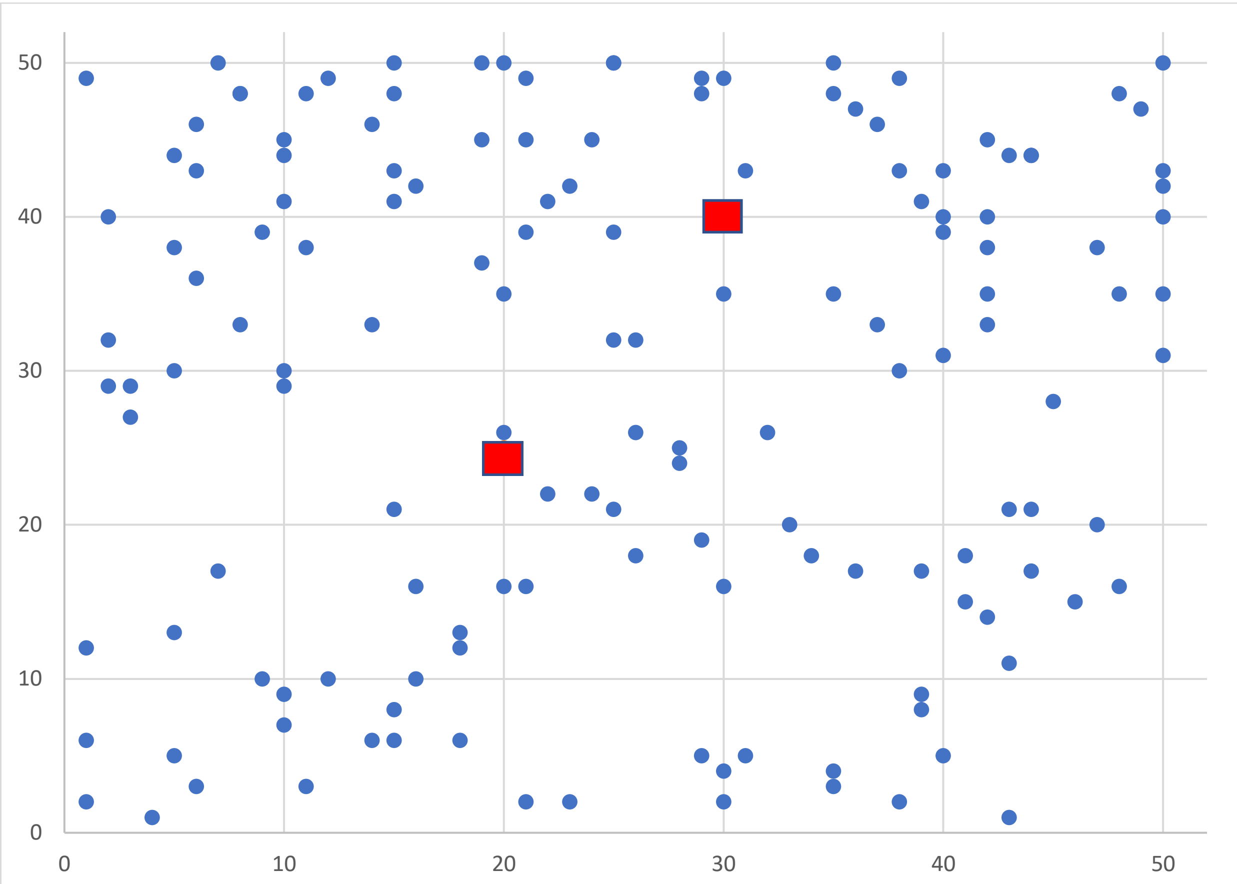

Figure 4.13. Customer location scatter plot

Table 4.14. Distance in miles for different algorithm combination

Algorithm run time limit	**0.5 sec**	**1 sec**	**2 sec**	**5 sec**
Algorithms	**Distance**	**Distance**	**Distance**	**Distance**
Path cheapest arc & Simulated annealing	1745.8	1746.8	1746.8	1746.8
Path cheapest arc & Tabu search	1705.3	1821.6	1838.7	1824.5
Path cheapest arc & Guided local search	1894.1	1890.6	1887.3	1790.5
Savings & Simulated annealing	1813.3	1813.3	1813.3	1813.3
Savings & Tabu search	1978.2	1842	1842.6	1816.3
Savings & Guided local search	1905.6	1832.3	1931.8	1731.6
Christofides & Simulated annealing	1819.2	1819.2	1819.2	1819.2
Christofides & Tabu search	1833.5	1882.4	1771	1940.1
Christofides & Guided local search	1751.3	1767.5	1857.2	1829.7

Table 4.15. Detailed results

Algorithm Combination	Algorithm Run time	Distance travelled Vehicle 1	Vehicle 2	Vehicle 3	Vehicle 4	Unsatisfied demand	Dropped visits
Path cheapest arc & Simulated annealing	0.5	421.8	425	422.1	476.9	244	32
	1	421.8	425	422.1	477.9	251	33
	2	421.8	425	422.1	477.9	251	33
	5	421.8	425	422.1	477.9	251	33
Path cheapest arc & Tabu search	0.5	414.7	430	441.5	419.1	343	48
	1	469.5	436.4	469.4	446.3	244	33
	2	465.8	443.3	452.1	477.5	258	34
	5	464.2	432.7	466.5	461.1	240	31
Path cheapest arc & Guided local search	0.5	484.4	504.2	475.5	430	225	30
	1	493.1	445.9	494.7	456.9	125	16
	2	443.2	462.1	501.1	480.9	195	26
	5	414.6	457	451.3	467.6	203	27
Savings & Simulated annealing	0.5	499.7	414.7	423.4	475.5	651	88
	1	499.7	414.7	423.4	475.5	651	88
	2	499.7	414.7	423.4	475.5	651	88
	5	499.7	414.7	423.4	475.5	651	88
Savings & Tabu search	0.5	482.1	520.6	458.9	516.6	540	71
	1	414.5	475.6	483.4	468.5	572	77
	2	451.7	419.9	536.3	434.7	583	73
	5	453	460.9	459.7	442.7	468	61
Savings & Guided local search	0.5	494.8	490.3	492.9	427.6	369	49
	1	441	436.3	503.4	451.6	379	48
	2	484.6	465.8	481.3	500.1	268	34
	5	420.7	463.5	428.3	419.1	316	42
Christofides & Simulated annealing	0.5	428.8	433.7	496.5	460.2	402	55
	1	428.8	433.7	496.5	460.2	402	55
	2	428.8	433.7	496.5	460.2	402	55
	5	428.8	433.7	496.5	460.2	402	55
Christofides & Tabu search	0.5	421.4	489.3	487.8	435	424	59
	1	537.9	453.6	464.1	426.8	424	58
	2	427.1	420.5	481.6	441.8	444	61
	5	448.7	508.8	529.2	453.4	334	47
Christofides & Guided local search	0.5	446.3	422.5	435.1	447.4	407	54
	1	423.9	448.6	450.3	444.7	305	41
	2	494.5	446.2	446.7	469.8	278	38
	5	425	448.4	523.6	432.7	277	39

When compared to the first case study, the second case study includes a greater number of customers that are being served by the same depots and vehicles. The second case study serves 148 customers using the 4 vehicles with capacities 30,20,30,20 respectively. This leads to some customer service requests to be dropped as seen in Table 15. The algorithm chooses the customers to be dropped by using a distance penalty value as depicted by equation (4.1). Each customer service request is assigned a distance penalty value that is directly proportional to the demand at that customer service location. Whenever a visit to a location is dropped, the penalty is added to the total distance traveled. The algorithm then finds a route that minimizes the total distance plus the sum of the penalties for all dropped locations.

$$Distance\ penalty = x * d_i \tag{4.1}$$

where

$x = constant$

$d_i = demand\ at\ node\ i$

For the purpose of this study, the value of x is chosen as 100. Based on Table 15, the shortest distance is produced by the combination of Path Cheapest Arc and Tabu Search with an algorithm run time limit of 1 second. This combination results in a total fleet distance of 1705.3 miles. Moreover, this combination serves 62.3% of the total demand and drops 48 customer locations which equals 343 unsatisfied units. However, this algorithm combination cannot be chosen as the best solely on the basis of distance traveled since satisfied and unsatisfied demand should also be taken into consideration while deciding the best combination. In these scenarios, a Cost-Benefit Analysis (CBA) helps to decide if a particular algorithm combination is likely to outweigh its drawbacks. Next section includes a CBA that helps to identify the best algorithm combination.

CHAPTER 5

RESULTS AND CONCLUSION

5.1 Case study 1

In this research, the objective is to minimize the total distance traveled by the vehicle fleet while catering to the predetermined and real-time customer service requests. The previous chapter discussed the performance of various algorithms and the nine combinations that were modeled using the 2-stage algorithm in Python. As seen from the results produced in the previous chapter different combinations produce different results and in some cases, the computational time had a significant influence on the quality of results. This section analyses the results obtained using the 2-stage algorithm for case studies one and two.

This section provides a brief analysis of the change in objective function values while using different methods in the 2-stage algorithm. Two scenarios were developed to analyze the case study, the first scenario only considered the three heuristic algorithms from the first stage and the second scenario used a combination of the first stage heuristic algorithms and local search metaheuristics from the second stage. Fig. 13 visualizes the values of the objective function for the first scenario where only the first stage heuristic was used. As seen in Fig. 13, Christofides algorithm produces the best result of 368 miles whereas Clark & Wright Savings algorithm produces the worst result of 462.9 miles, which is approximately 25% greater than the best result. Moreover, Path Cheapest Arc algorithm produces a slightly worse result of 381 miles, which is approximately 4% greater than the best result. Interestingly, Path Cheapest Arc, which is a simple rule of thumb method that iteratively extends the current route by choosing the closest node could produce a much better result than the widely popular Clark & Wright Savings heuristics. It even performs almost on par with the Christofides algorithm in this specific case study.

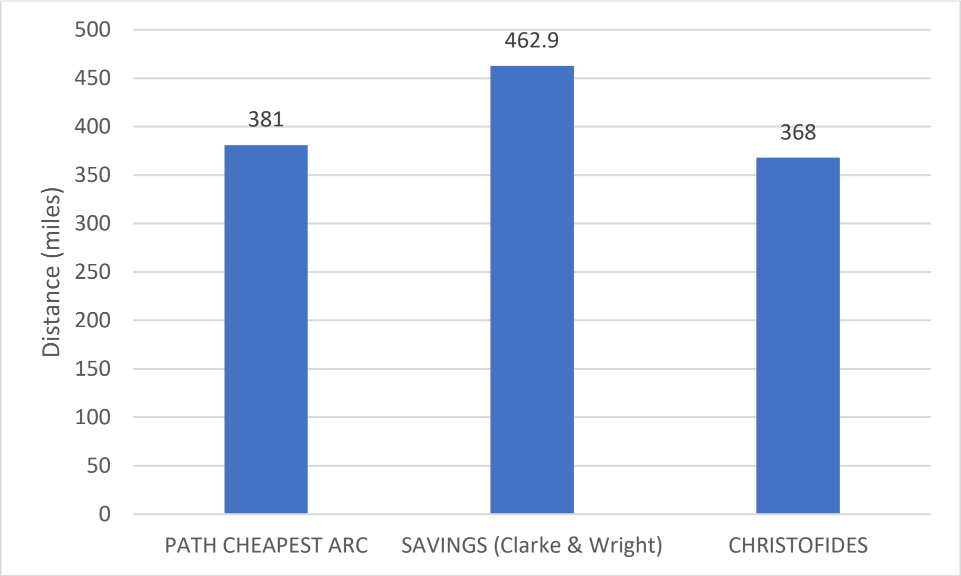

Figure 5.1. Scenario 1 results

In the case of scenario 2, the results for nine combinations of algorithms have been presented in Fig. 5.1, and Fig. 5.2. As seen below, Fig. 5.2 shows the best result from among the different computational time for each of the nine combinations.

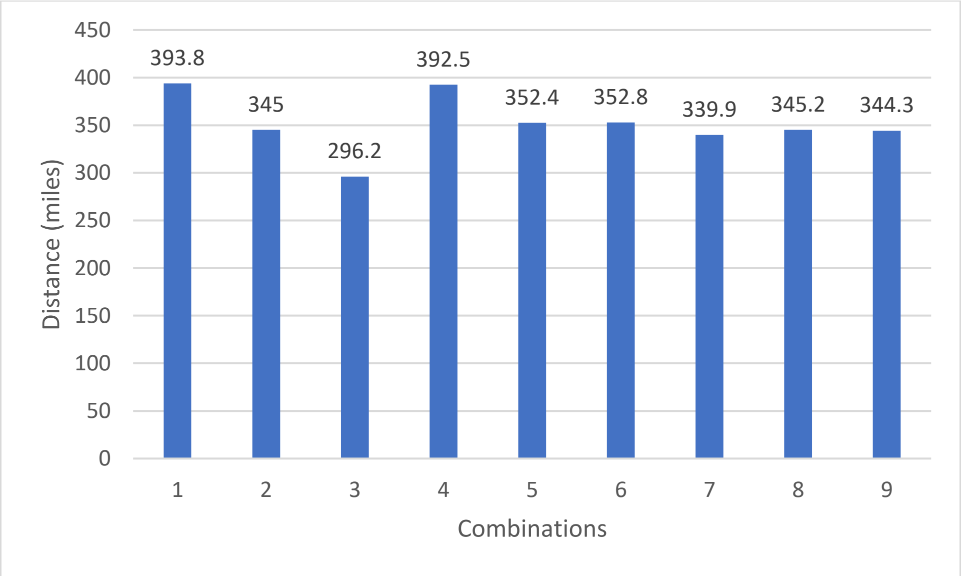

Figure 5.2. Total distance comparison

The best result has been produced by the third combination which used Path Cheapest Arc method in the first stage and Guided Local Search in the second stage. It produced a result of 296.2 miles while running the algorithm for 1 second.

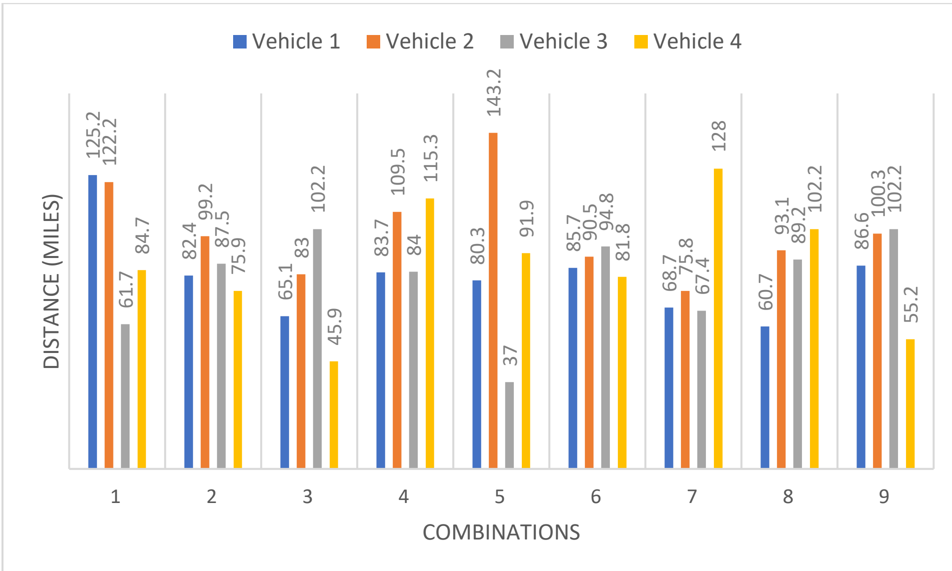

Figure 5.3. Best results for each combination classified by vehicle

Fig. 5.3 represents the best results of all the nine combinations by depicting the distance traveled by each vehicle. Looking at combination 3 results, vehicle 3 traversed the maximum distance of 102.2 miles, whereas vehicle 4 traversed the minimum distance of 45.9 miles. These results are not surprising as vehicle 3 has 50% more capacity than vehicle 4, so vehicle three automatically will cater to more customer requests. The worst result of 393.8 miles (33% higher than the best result) is produced by the combination 1 that uses Path Cheapest Arc algorithm and Simulated Annealing. For this combination, the second stage worsened the result produced by the first stage, and the possible reason might be in the Simulated Annealing getting caught in a local optima. Also, Christofides algorithm in the first stage tends to produce almost identical results whereas those combinations that use Path Cheapest Arc method tends to show the highest fluctuation. Here are the routes visualised for combination three which produced the best result.

Fig. 5.4 shows the routes produced by the 2-stage algorithm for all the customer requests available at the start of the planning period. All the orange squares marks the customer locations with their location coordinates in the Euclidean space provided in square brackets. The black squares are the depots where the vehicle routes start and ends. Each coloured arrow lines represent each of the four vehicles. For the below routes, only three vehicles have been utilized.

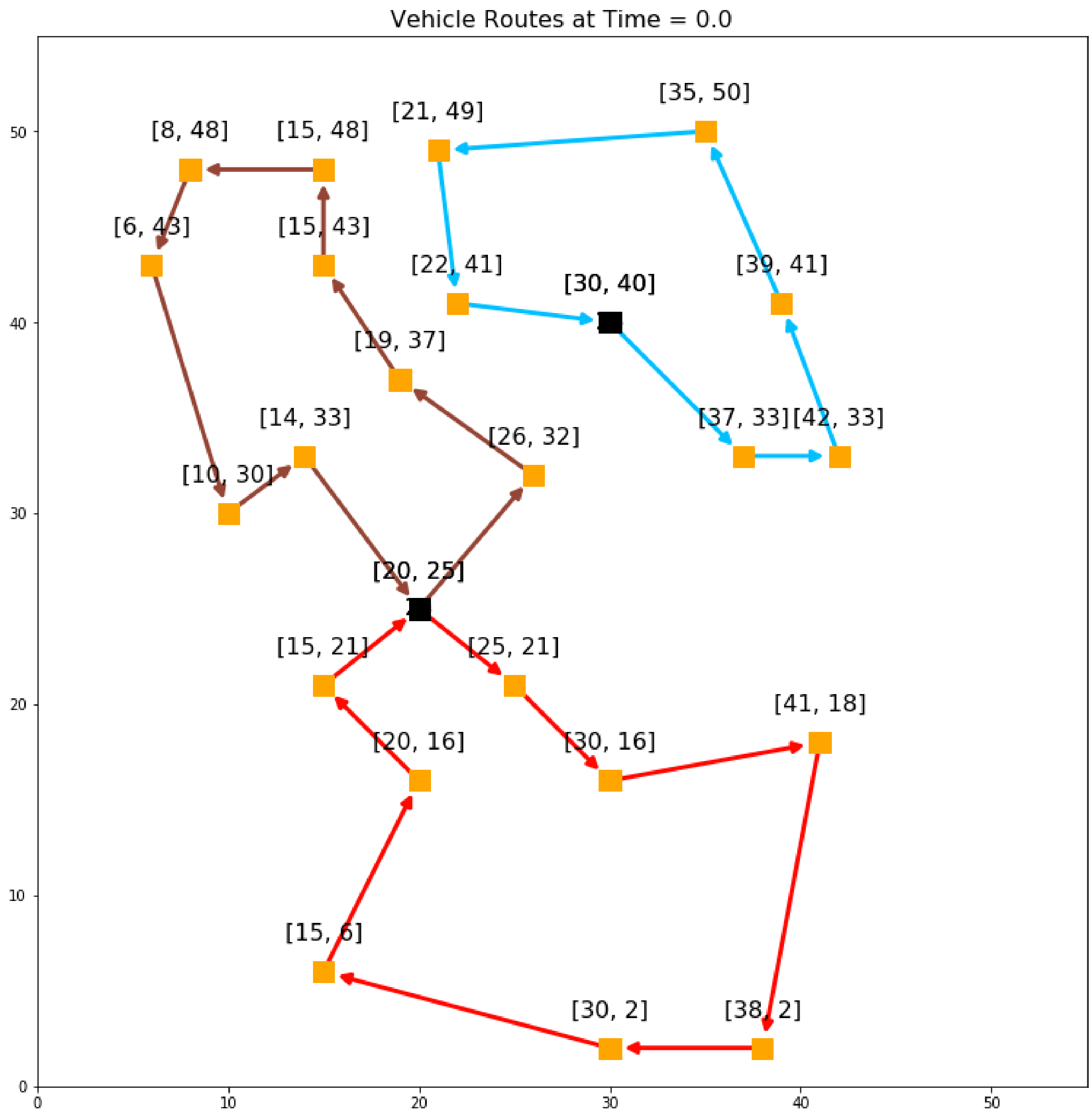

Figure. 5.4. Vehicle routes at the beginning of the planning period

Gradually, with the passage of time, new customer requests starts to appear when the vehicles are already in transit. Fig. 5.5 shows the new routes developed at the 13th minute when a new customer request came in with a pickup request of 3 units at the location [5,5]. The green squares represent the instantaneous vehicle locations with location coordinates at the specified time. Also, all the vehicles have been utilized under this scenario, whereas only 3 vehicles were utilized when the routes were planned at the beginning of the planning period.

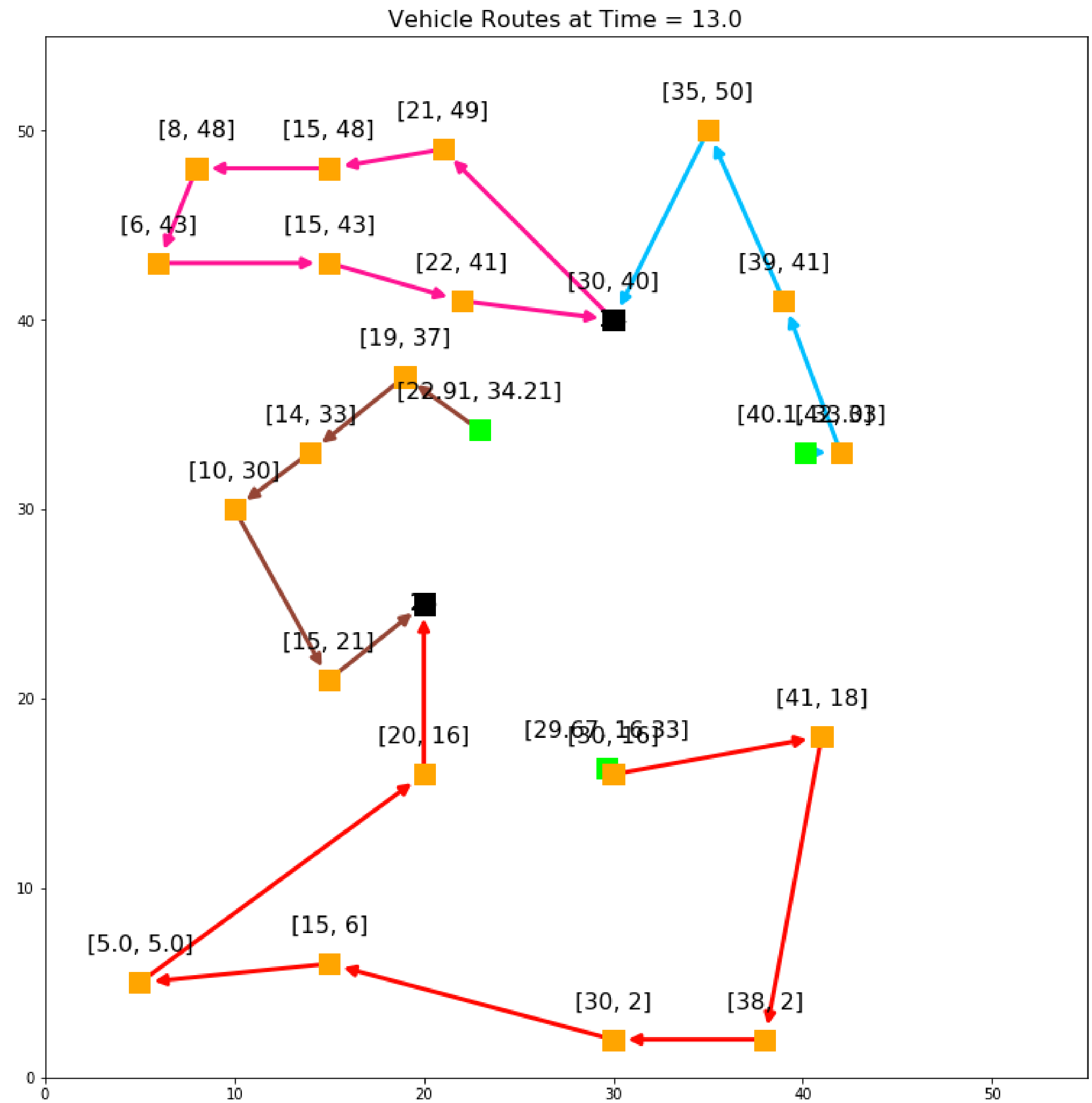

Figure 5.5. Vehicle routes at the 13th minute

Then at the 31st minute, a new pickup request for 4 quantities arrives at the location [5, 38]. The existing routes produced at the 13th minute is modified to include the new request as seen in Fig. 5.6 below. No major re-routing has happened except for the vehicle serving the new customer location.

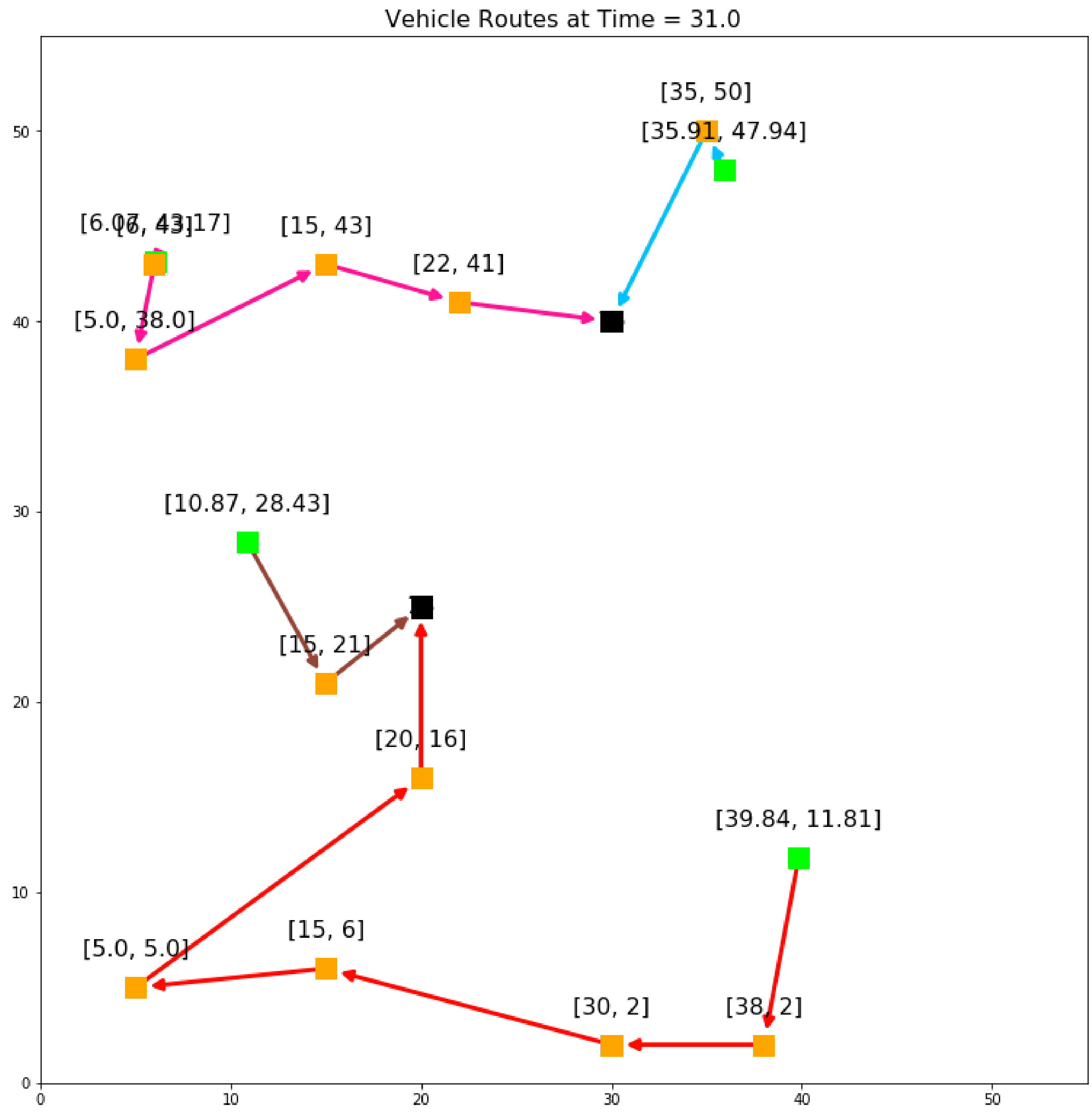

Figure 5.6. Vehicle route at the 31st minute

At the 45th minute, another new customer pickup request comes in with a pickup quantity of 1 unit at the location [19, 45]. Again, no major re-routing has been done except for the vehicle that serves the new location as seen in Fig. 5.7. The vehicle that was represented using the blue arrows have finished serving customers and is at the depot marked by location [30,40].

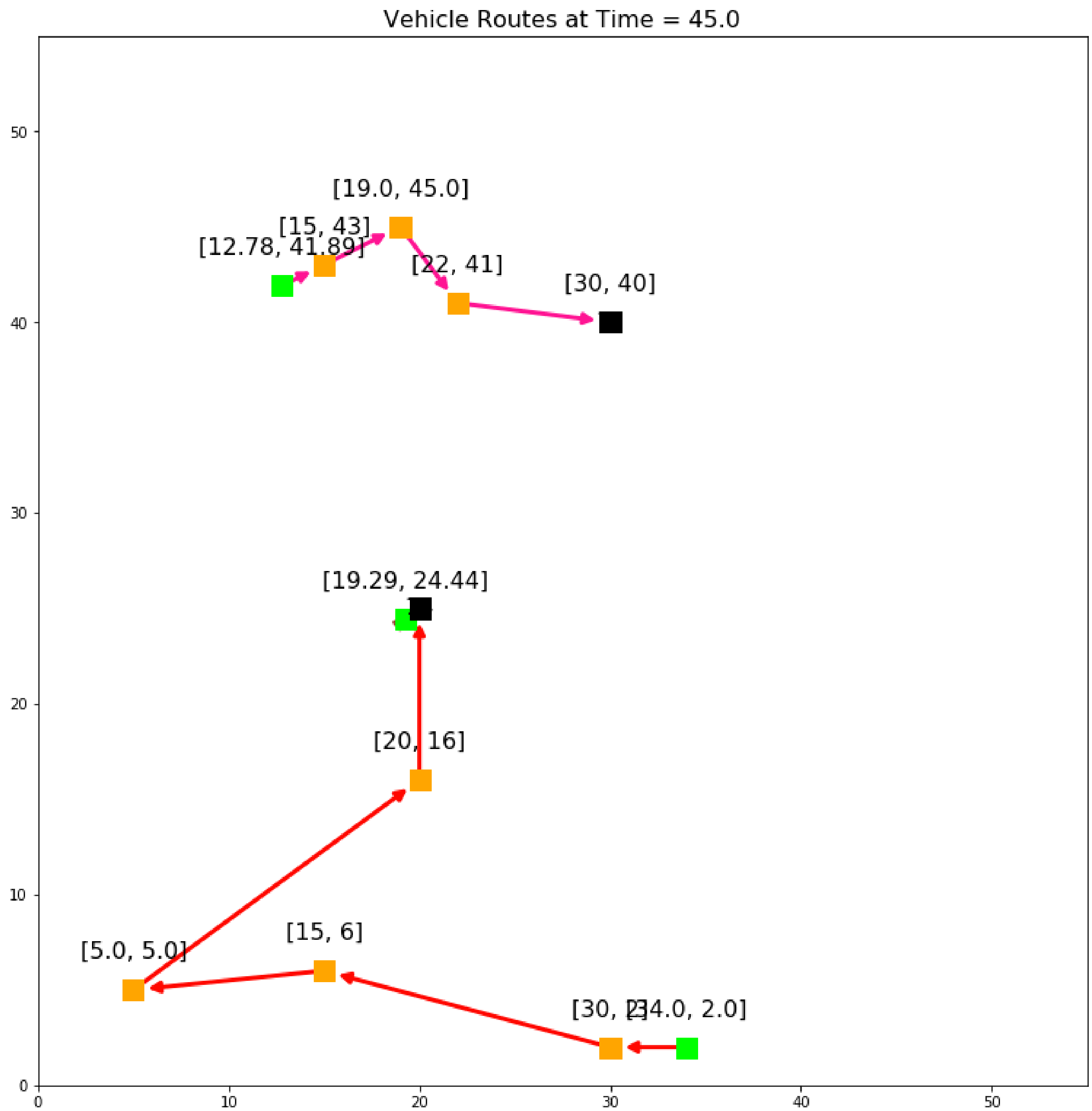

Figure 5.7. Vehicle route at the 45th minute

A new customer pickup request of 6 units arrived in the 51st minute at the location [45, 28]. In this scenario, the new routes generated in Fig. 5.8 shows that the vehicle that was previously idle has been assigned to serve the new customer. The vehicle represented by the brown arrow has finished serving all the customers and is idling at the depot marked by the location [20,25].

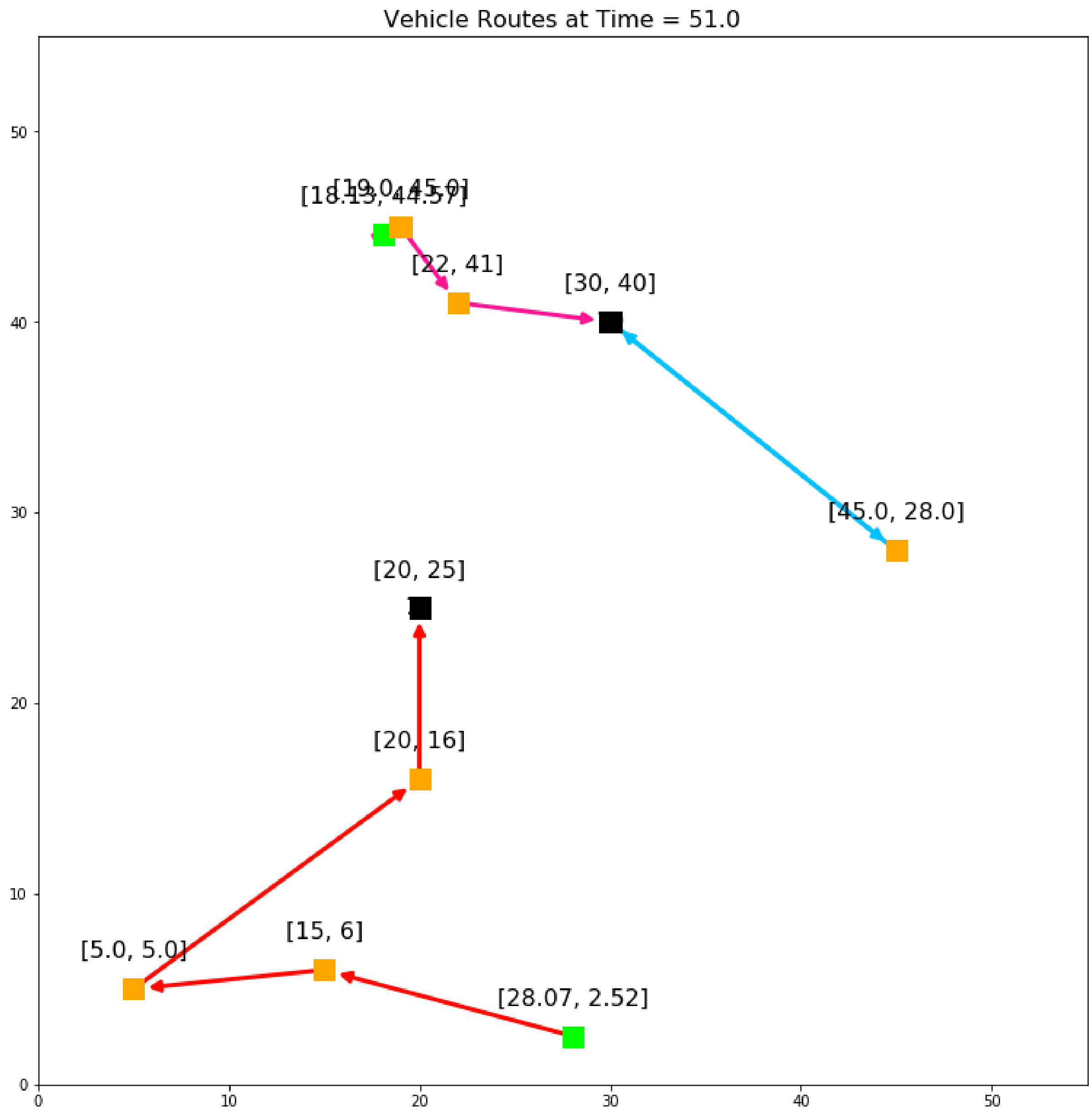

Figure 5.8. Vehicle route at the 51st minute

Finally, in the 66th minute, the last customer request arrives with a pickup quantity of 4 units at the location [35, 35]. As seen in Fig. 5.9 below, the request is served by the vehicle that is represented using the blue arrow lines. Vehicles represented by the pink and the brown arrow line are idling at the depot after serving all the customers. At the end of the time period, all the vehicles have travelled a combined distance of 296.2 miles.

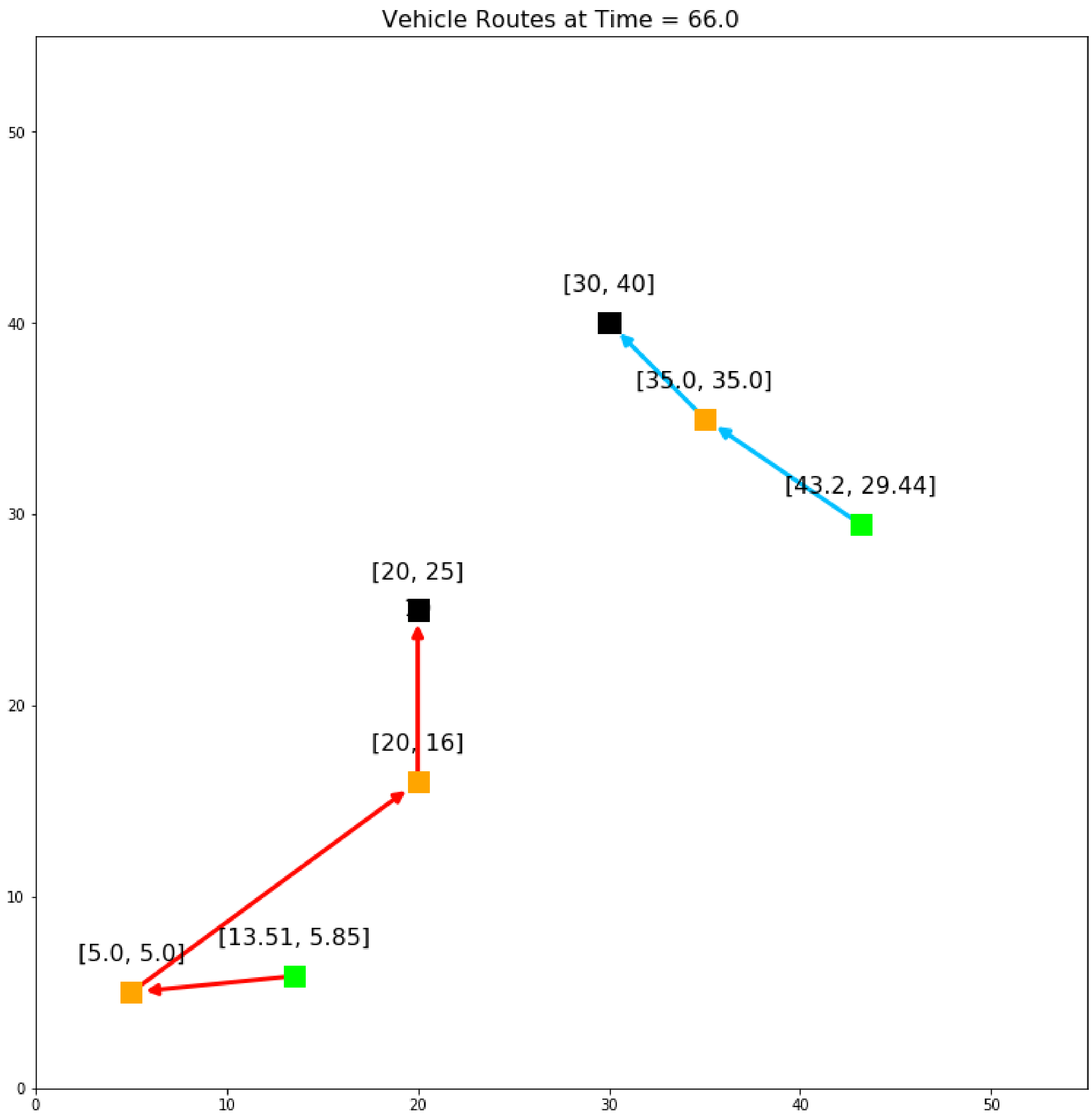

Figure 5.9. Vehicle route at the 66th minute

Table 5.1 shows the summary of the D-VRP problem solved using the combination of Path Cheapest Arc algorithm and Guided Local Search algorithm.

Table 5.1. Route summary

Depot	Vehicle	Customers Served	Total Quantity	Distance Travelled
1	1 [Pink lines]	3,16,14,2,24,18,25,12	28	65.1 miles
	2 [Blue lines]	11,19,7,21,27,26	23	83 miles
2	3 [Red lines]	9,10,6,20,13,15,23,4	28	102.2 miles
	4 [Brown lines]	1,17,8,22,5	16	45.9 miles

Vehicle 2 has picked up 23 units while its capacity is 20 units. The reason why it picked up more units than its capacity is because vehicle 2 can be seen idling at the depot while new routes were designed at the 45th minute. Whenever a vehicle visits a depot, it unloads all the units that it carried and starts again with full capacity.

5.2 Case Study 2 results

This research aims to minimize the total distance traveled by the vehicle fleet while catering to the predetermined and real-time customer service requests. The previous chapter discussed various case studies used to assess the performance of the 2-stage decision algorithm. This section analyses the results of the second case study presented in the previous chapter.

5.2. Cost analysis

Table 17 tabulates the various costs including expense, revenue, profit, lost opportunity cost, and adjusted profit for each algorithm combinations using equations 5.1, 5.2, 5.3, 5.4, and 5.5.

$$Expense = \sum_{i=1}^{n}(C_i + T_i)D_i \tag{5.1}$$

$$Revenue = S * \sum_{j=1}^{148} Q_j \tag{5.2}$$

$$Profit = Revenue - Expense \tag{5.3}$$

$$Lost\ opportunity\ cost = \sum_{j=1}^{148}[\left(S * q_j\right) - (d_{jk} * 2 * \frac{(C_{1k}+C_{2k})}{2})] \qquad (k = 1,2) \tag{5.4}$$

$$Adjusted\ profit = Profit - Lost\ opportunity\ cost \tag{5.5}$$

where

$C_i = Cost\ per\ mile\ in\ dollars\ for\ vehivle\ i$

$T_i = \begin{cases} 0.1\ if\ vehicle\ i\ is\ fitted\ with\ a\ trailer \\ 0\ otherwise \end{cases}$

$D_i = Distance\ travelled\ by\ vehicle\ i$

$S = service\ fee\ in\ dollars\ obtained\ for\ each\ unit\ collected$

$Q_j = number\ of\ units\ collected\ from\ node\ j$

$q_j = number\ of\ units\ at\ node\ j$

$d_{jk} = distance\ from\ node\ j\ to\ the\ closest\ depot\ k$

For the purpose of this study, each unit of customer demand served is assumed to earn $15 for the business, which is denoted by S. The cost per mile for vehicles has been assumed based on the data from American Transportation Research Institute (ATRI). The latest data from ATRI shows that the average trucking cost per mile for straight trucks in the U.S. is $1.63. For the purpose of this study, vehicles 2 and 4 with 20-unit capacity are considered to have a cost of $1.55 per mile and, vehicles 1 and 3 with 30-unit capacity are considered to have a cost of $1.63 per mile. Adding extra capacity by attaching a trailer to an existing vehicle increases the cost per mile by $0.1.

The profit shown in table 5.2 is calculated based on the expense and revenue in a certain time frame. However, it does not take into consideration the service level or the number of unserved customer locations. This issue is solved by introducing the lost opportunity cost for each unserved customer location. The lost opportunity cost is the revenue lost due to not serving a customer location. Subtracting the lost opportunity cost from the actual profit gives the adjusted profit, which can be used to compare algorithm combinations while taking into account the distance traveled as well as the service level.

Based on the values of adjusted profit, it can be observed that the Path Cheapest Arc and Guided Local Search algorithm combination produced the best adjusted profit value of $8029.35 for a run time limit of 1 second. This algorithm combination produced a total fleet distance of 1890.6 miles and satisfied 86% of the customer demand. However, this combination dropped 16 customer requests, which equal 125 units of unsatisfied demand. The lowest adjusted profit is produced by the combination of Savings and Simulated annealing algorithm, which produced a negative value of $3821.75. This is due to the high lost opportunity cost and comparatively higher fleet distance. All negative values are placed in Parenthesis in Table 5.2.

Table 5.2. Cost Summary

Algorithm Combination	Algorithm Run time	Expense	Revenue	Profit	Lost opportunity	Adjusted profit	Service level
Path cheapest arc & Simulated annealing	0.5	$2,773.50	$ 9,975.00	$7,201.50	$ 1,694.70	$ 5,506.80	73%
	1	$2,775.05	$ 9,870.00	$7,094.95	$ 1,749.80	$ 5,345.15	72%
	2	$2,775.05	$ 9,870.00	$7,094.95	$ 1,749.80	$ 5,345.15	72%
	5	$2,775.05	$ 9,870.00	$7,094.95	$ 1,749.80	$ 5,345.15	72%
Path cheapest arc & Tabu search	0.5	$2,711.71	$ 8,490.00	$5,778.29	$ 2,477.20	$ 3,301.09	62%
	1	$2,898.59	$ 9,975.00	$7,076.41	$ 1,350.90	$ 5,725.51	73%
	2	$2,923.42	$ 9,765.00	$6,841.58	$ 1,553.10	$ 5,288.48	72%
	5	$2,902.43	$10,035.00	$7,132.57	$ 1,512.80	$ 5,619.77	74%
Path cheapest arc & Guided local search	0.5	$3,012.65	$10,260.00	$7,247.35	$ 1,417.40	$ 5,829.95	75%
	1	$3,009.45	$11,760.00	$8,750.55	$ 721.20	$ 8,029.35	86%
	2	$3,000.86	$10,710.00	$7,709.14	$ 1,198.00	$ 6,511.14	79%
	5	$2,844.55	$10,590.00	$7,745.45	$ 1,427.90	$ 6,317.55	78%
Savings & Simulated annealing	0.5	$2,884.46	$ 3,870.00	$ 985.54	$ 4,807.29	$(3,821.75)	28%
	1	$2,884.46	$ 3,870.00	$ 985.54	$ 4,807.29	$(3,821.75)	28%
	2	$2,884.46	$ 3,870.00	$ 985.54	$ 4,807.29	$(3,821.75)	28%
	5	$2,884.46	$ 3,870.00	$ 985.54	$ 4,807.29	$(3,821.75)	28%
Savings & Tabu search	0.5	$3,141.49	$ 5,535.00	$2,393.51	$ 4,037.80	$(1,644.29)	41%
	1	$2,926.93	$ 5,055.00	$2,128.07	$ 4,126.40	$(1,998.33)	37%
	2	$2,935.07	$ 4,890.00	$1,954.93	$ 4,632.70	$(2,677.77)	36%
	5	$2,888.28	$ 6,615.00	$3,726.72	$ 3,564.90	$ 161.82	49%
Savings & Guided local search	0.5	$3,032.70	$ 8,100.00	$5,067.30	$ 2,513.10	$ 2,554.20	59%
	1	$2,915.62	$ 7,950.00	$5,034.38	$ 2,720.90	$ 2,313.48	58%
	2	$3,071.56	$ 9,615.00	$6,543.44	$ 1,895.30	$ 4,648.14	71%
	5	$2,751.90	$ 8,895.00	$6,143.10	$ 2,411.60	$ 3,731.50	65%
Christofides & Simulated annealing	0.5	$2,893.78	$ 7,605.00	$4,711.22	$ 2,684.50	$ 2,026.72	56%
	1	$2,893.78	$ 7,605.00	$4,711.22	$ 2,684.50	$ 2,026.72	56%
	2	$2,893.78	$ 7,605.00	$4,711.22	$ 2,684.50	$ 2,026.72	56%
	5	$2,893.78	$ 7,605.00	$4,711.22	$ 2,684.50	$ 2,026.72	56%
Christofides & Tabu search	0.5	$2,914.66	$ 7,275.00	$4,360.34	$ 2,920.20	$ 1,440.14	53%
	1	$2,997.88	$ 7,275.00	$4,277.12	$ 2,698.20	$ 1,578.92	53%
	2	$2,817.75	$ 6,975.00	$4,157.25	$ 3,209.80	$ 947.45	51%
	5	$3,085.39	$ 8,625.00	$5,539.61	$ 2,234.40	$ 3,305.21	63%
Christofides & Guided local search	0.5	$2,785.03	$ 7,530.00	$4,744.97	$ 2,822.90	$ 1,922.07	55%
	1	$2,809.56	$ 9,060.00	$6,250.44	$ 2,062.30	$ 4,188.14	66%
	2	$2,953.96	$ 9,465.00	$6,511.04	$ 1,760.20	$ 4,750.84	69%
	5	$2,911.92	$ 9,480.00	$6,568.08	$ 1,683.60	$ 4,884.48	70%

With the exception of algorithm combinations including Simulated annealing algorithm, all the algorithm combinations produced varying results based on the algorithm run time. Simulated annealing might get trapped in a local optima, and that might be the reason why the combinations produced the same results for different algorithm run time.

Here is the result summary for the combination Path Cheapest Arc and Guided Local Search with an algorithm run time limit of 2 seconds, which produced the best result. The results of the algorithm are shown in the below table, and fig 5.10 shows the distance traveled by each vehicle while running the 2-stage decision algorithm.

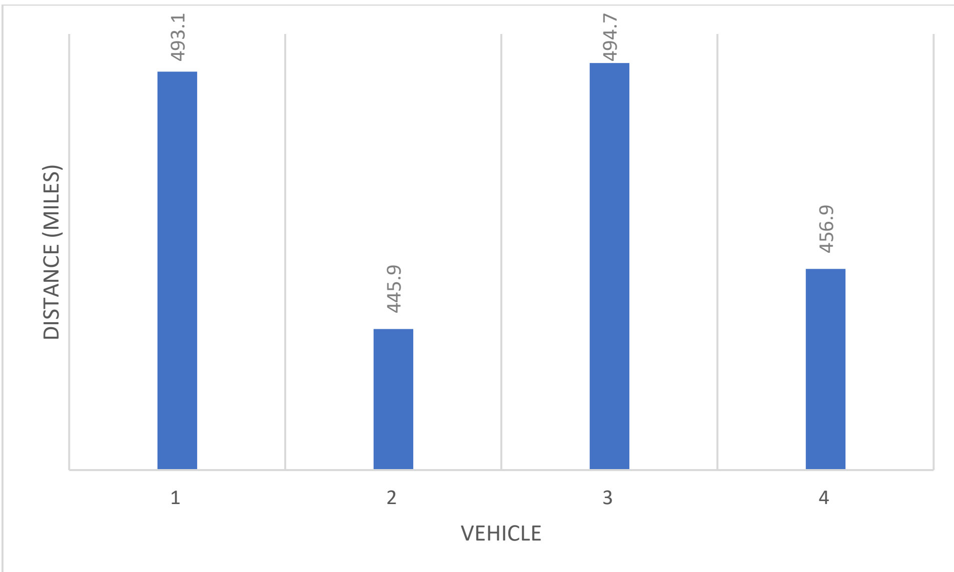

Figure 5.10. Distance traveled by each vehicle

As seen in fig. 5.10, Vehicle 3 with capacity 30 traveled the highest distance of 494.7 miles while vehicle 2 with capacity 20 traveled the least distance of 445.9 miles.

Table 5.3. Route summary

Number of Customers	**Total Demand**	**Distance traveled by vehicle fleet**	**Number of dropped visits**	**Unsatisfied Demand**	**Service Level**
148	909	1890.6	26	125	86%

The vehicles traversed a total distance of 1890.6 miles while serving 122 customers with a service level of 86% and dropping 26 customer visits based on the distance penalty assigned to each of the customer locations. Good customer service can be the difference between being able to compete and survive and failing for businesses.

Acquiring new customers is important, but retaining existing customers accelerates profitable growth since its much cheaper to retain existing customers, and the crucial factor in keeping customers is the level of customer service the business provides. So the aim of every successful business is to serve more number of customers, at the minimum cost. In this study, in order to serve more customers and improve the service level, various potential decisions and their results are discussed in the following section. Table 16 shows the potential decisions that could be implemented to improve the service level.

Table 5.4. Decision table

Decision Number	Decision
1	Increasing the vehicle capacity
2	Adding an extra vehicle

The following section discusses the various scenarios that occur when implementing each of these decisions. All scenarios use the algorithm combination of Path Cheapest Arc and Guided Local Search with an algorithm run time limit of 2 seconds as it produced the best result in the previous section.

5.2.2 Decision 1 - Increasing the capacity of the vehicles

The first potential decision on table 5.4 is to increase the existing vehicle capacity of the vehicles at each depot. This option provides a viable way to serve more customers at a low overhead cost. For the purpose of this study, it assumes that only one vehicle is being modified to increase the existing capacity at a time. This section discusses and tabulates the results for the four scenarios that arise when implementing the decision to increase the capacity of the vehicles by attaching a trailer with capacity 20.

5.2.2.1 Increasing capacity of vehicle 1

Table 5.5. Route summary

Algorithm Combination	Algorithm Run time	Distance travelled (miles)					Unsatisfied demand	Dropped visits
		Vehicle 1	Vehicle 2	Vehicle 3	Vehicle 4	Total		
Path cheapest arc & Simulated annealing	.5 sec	486.1	443.7	477.9	444.5	1852.2	271	36
	1 sec	486.1	443.7	477.9	444.5	1852.2	271	36
	2 sec	486.1	443.7	477.9	444.5	1852.2	271	36
	5 sec	486.1	443.7	477.9	444.5	1852.2	271	36
Path cheapest arc & Tabu search	.5 sec	442	512.7	469.5	447.8	1872	259	35
	1 sec	487.3	433.1	438.2	418	1776.6	239	31
	2 sec	475.4	424	431.6	429.5	1760.5	259	34
	5 sec	454.4	474.8	458.7	433.1	1821	237	32
Path cheapest arc & Guided local search	.5 sec	495.9	433.4	444.2	470.8	1844.3	171	23
	1 sec	494.7	425.4	421.5	435.7	1777.3	202	28
	2 sec	421.9	472.3	465	430.7	1789.9	207	29
	5 sec	427.6	461.4	452.2	449.7	1790.9	208	30
Savings & Simulated annealing	.5 sec	454	487.6	479.3	447.2	1868.1	653	89
	1 sec	454	487.6	479.3	447.2	1868.1	653	89
	2 sec	454	487.6	479.3	447.2	1868.1	653	89
	5 sec	454	487.6	479.3	447.2	1868.1	653	89
Savings & Tabu search	.5 sec	468.3	416.7	489.8	414.3	1789.1	546	72
	1 sec	451.4	465	497.8	464.2	1878.4	496	66
	2 sec	415.5	419.8	482.8	465.2	1783.3	508	66
	5 sec	507.1	422	435.4	474.3	1838.8	379	50
Savings & Guided local search	.5 sec	496	429.2	464.3	524	1913.5	419	54
	1 sec	479.5	475.6	461.5	463.5	1880.1	329	42
	2 sec	451.8	462.8	462.9	457.7	1835.2	310	40
	5 sec	450.4	429.9	421.1	439.3	1740.7	346	44
Christofides & Simulated annealing	.5 sec	483.1	466.9	456.6	414.2	1820.8	351	47
	1 sec	483.1	466.9	456.6	414.2	1820.8	351	47
	2 sec	483.1	466.9	456.6	414.2	1820.8	351	47
	5 sec	483.1	466.9	456.6	414.2	1820.8	351	47
Christofides & Tabu search	.5 sec	433.8	447.2	471.6	478.6	1831.2	398	56
	1 sec	480	469.8	453.6	493.4	1896.8	399	54
	2 sec	423.9	442.9	509.8	446.6	1823.2	432	55
	5 sec	466.5	443.3	417.7	488.5	1816	413	54
Christofides & Guided local search	.5 sec	447.5	458.6	482.4	416.5	1805	276	39
	1 sec	521	419.6	457.1	428.5	1826.2	360	49
	2 sec	428.7	432.3	477.9	417.2	1756.1	269	39
	5 sec	416.6	455.4	449.3	457.8	1779.1	269	38

The combination of Savings and Guided local search with a run time limit of 5 seconds produced the lowest total fleet distance of 1740.7 miles. However, it dropped 44 customers, which equals 346 total unsatisfied quantities as seen in Table 5.5. The below table explains the cost-benefit analysis done after increasing the capacity of vehicle by 20 units.

Table 5.6. Cost summary

Algorithm Combination	Algorithm Run time	Expense	Revenue	Profit	Lost opportunity	Adjusted profit	Service level
Path cheapest arc & Simulated annealing	.5 sec	$2,996.64	$ 9,570.00	$6,573.36	$ 1,797.90	$ 4,775.46	70%
	1 sec	$2,996.64	$ 9,570.00	$6,573.36	$ 1,797.90	$ 4,775.46	70%
	2 sec	$2,996.64	$ 9,570.00	$6,573.36	$ 1,797.90	$ 4,775.46	70%
	5 sec	$2,996.64	$ 9,570.00	$6,573.36	$ 1,797.90	$ 4,775.46	70%
Path cheapest arc & Tabu search	.5 sec	$3,018.72	$ 9,750.00	$6,731.28	$ 1,576.60	$ 5,154.68	72%
	1 sec	$2,876.50	$10,050.00	$7,173.50	$ 1,719.90	$ 5,453.60	74%
	2 sec	$2,848.88	$ 9,750.00	$6,901.13	$ 1,559.10	$ 5,342.03	72%
	5 sec	$2,941.04	$10,080.00	$7,138.96	$ 1,474.90	$ 5,664.06	74%
Path cheapest arc & Guided local search	.5 sec	$2,983.46	$11,070.00	$8,086.54	$ 1,081.30	$ 7,005.24	81%
	1 sec	$2,877.58	$10,605.00	$7,727.42	$ 1,282.40	$ 6,445.02	78%
	2 sec	$2,887.49	$10,530.00	$7,642.51	$ 1,232.60	$ 6,409.91	77%
	5 sec	$2,889.04	$10,515.00	$7,625.96	$ 1,363.00	$ 6,262.96	77%
Savings & Simulated annealing	.5 sec	$3,015.62	$ 3,840.00	$ 824.38	$ 4,947.90	$(4,123.52)	28%
	1 sec	$3,015.62	$ 3,840.00	$ 824.38	$ 4,947.90	$(4,123.52)	28%
	2 sec	$3,015.62	$ 3,840.00	$ 824.38	$ 4,947.90	$(4,123.52)	28%
	5 sec	$3,015.62	$ 3,840.00	$ 824.38	$ 4,947.90	$(4,123.52)	28%
Savings & Tabu search	.5 sec	$2,896.58	$ 5,445.00	$2,548.42	$ 4,028.90	$(1,480.48)	40%
	1 sec	$3,032.60	$ 6,195.00	$3,162.40	$ 3,469.40	$ (307.00)	45%
	2 sec	$2,877.53	$ 6,015.00	$3,137.47	$ 3,461.70	$ (324.23)	44%
	5 sec	$2,976.25	$ 7,950.00	$4,973.75	$ 2,696.70	$ 2,277.05	58%
Savings & Guided local search	.5 sec	$3,092.35	$ 7,350.00	$4,257.65	$ 3,209.90	$ 1,047.75	54%
	1 sec	$3,037.39	$ 8,700.00	$5,662.62	$ 2,314.20	$ 3,348.42	64%
	2 sec	$2,962.92	$ 8,985.00	$6,022.08	$ 2,235.70	$ 3,786.38	66%
	5 sec	$2,812.85	$ 8,445.00	$5,632.16	$ 2,678.60	$ 2,953.56	62%
Christofides & Simulated annealing	.5 sec	$2,945.73	$ 8,370.00	$5,424.27	$ 2,325.00	$ 3,099.27	61%
	1 sec	$2,945.73	$ 8,370.00	$5,424.27	$ 2,325.00	$ 3,099.27	61%
	2 sec	$2,945.73	$ 8,370.00	$5,424.27	$ 2,325.00	$ 3,099.27	61%
	5 sec	$2,945.73	$ 8,370.00	$5,424.27	$ 2,325.00	$ 3,099.27	61%
Christofides & Tabu search	.5 sec	$2,954.17	$ 7,665.00	$4,710.83	$ 2,730.10	$ 1,980.73	56%
	1 sec	$3,062.73	$ 7,650.00	$4,587.27	$ 2,898.50	$ 1,688.77	56%
	2 sec	$2,943.05	$ 7,155.00	$4,211.95	$ 3,050.70	$ 1,161.25	52%
	5 sec	$2,932.19	$ 7,440.00	$4,507.81	$ 2,982.10	$ 1,525.71	55%
Christofides & Guided local search	.5 sec	$2,916.89	$ 9,495.00	$6,578.11	$ 1,641.00	$ 4,937.11	70%
	1 sec	$2,960.96	$ 8,235.00	$5,274.04	$ 2,513.70	$ 2,760.34	60%
	2 sec	$2,837.35	$ 9,600.00	$6,762.65	$ 1,726.20	$ 5,036.45	70%
	5 sec	$2,868.54	$ 9,600.00	$6,731.46	$ 1,679.00	$ 5,052.46	70%

Table 5.6 shows the various costs for increasing the capacity of vehicle 1 by 20 units. Comparing various adjusted profit for different algorithm combinations, Pathcheapest Arc and Guided local search algorithm with a run limit of 0.5 seconds produced the highest adjusted profit of $7005.24, while serving 81% of the customers.

5.2.2.2 Increasing capacity of vehicle 2

Table 5.7. Route summary

Algorithm Combination	Algorithm Run time	Distance travelled (miles)					Unsatisfied demand	Dropped visits
		Vehicle 1	Vehicle 2	Vehicle 3	Vehicle 4	Total		
Path cheapest arc & Simulated annealing	.5 sec	482.5	429.4	508.2	466.4	1886.5	233	29
	1 sec	482.5	429.4	508.2	466.4	1886.5	233	29
	2 sec	482.5	429.4	508.2	466.4	1886.5	233	29
	5 sec	482.5	429.4	508.2	466.4	1886.5	233	29
Path cheapest arc & Tabu search	.5 sec	442.4	429.5	474.4	451.6	1797.9	238	31
	1 sec	433.4	438.6	446.1	474.3	1792.4	229	31
	2 sec	415.7	521	471.3	471.7	1879.7	198	24
	5 sec	473.9	422.5	442.7	460.7	1799.8	205	27
Path cheapest arc & Guided local search	.5 sec	463.2	456.5	434.3	444.5	1798.5	212	29
	1 sec	509.1	436.8	427.5	454.3	1827.7	208	27
	2 sec	440.2	492.6	427.7	438.3	1798.8	207	27
	5 sec	448.5	466	420.9	425.3	1760.7	210	29
Savings & Simulated annealing	.5 sec	520.5	486.7	444.5	459.2	1910.9	495	66
	1 sec	520.5	486.7	444.5	459.2	1910.9	495	66
	2 sec	520.5	486.7	444.5	459.2	1910.9	495	66
	5 sec	520.5	486.7	444.5	459.2	1910.9	495	66
Savings & Tabu search	.5 sec	473	496.9	456.2	465.2	1891.3	633	86
	1 sec	463.5	419.3	511.3	460.5	1854.6	461	57
	2 sec	478.1	511.1	436.5	466.5	1892.2	421	53
	5 sec	420.1	457.7	502.4	424.2	1804.4	414	54
Savings & Guided local search	.5 sec	541.3	475.1	491.1	443.4	1950.9	425	54
	1 sec	445.9	473.6	467.3	461.9	1848.7	367	49
	2 sec	469.2	469.6	445.1	433	1816.9	286	37
	5 sec	478.9	464.2	472.9	441.8	1857.8	256	30
Christofides & Simulated annealing	.5 sec	492	439.7	501.8	482.9	1916.4	387	53
	1 sec	492	439.7	501.8	482.9	1916.4	387	53
	2 sec	492	439.7	501.8	482.9	1916.4	387	53
	5 sec	492	439.7	501.8	482.9	1916.4	387	53
Christofides & Tabu search	.5 sec	462.5	505.3	416.1	455.4	1839.3	364	49
	1 sec	435.6	467.2	444.3	451.6	1798.7	274	37
	2 sec	481.5	460.1	466	437.8	1845.4	313	43
	5 sec	477.6	509	431.2	446.6	1864.4	309	43
Christofides & Guided local search	.5 sec	487.3	483	478.7	443.7	1892.7	266	35
	1 sec	428.8	450.8	420.5	444.3	1744.4	343	48
	2 sec	486.5	439.9	429.6	456	1812	252	35
	5 sec	450.6	493	520.2	426.1	1889.9	242	33

Table 5.7 shows the route summary for increasing the capacity of vehicle 2 by 20 units. In this scenario, the combination of Christofides and Guided local search with a run time of 1 second produces the lowest distance of 1744.4 miles while serving 100 customer locations. The below table shows the various costs associated with implementing this decision.

Table 5.8. Cost summary

Algorithm Combination	Algorithm Run time	Expense	Revenue	Profit	Lost opportunity	Adjusted profit	Service level
Path cheapest arc & Simulated annealing	.5 sec	$3,046.27	$10,140.00	$7,093.73	$ 1,521.30	$ 5,572.43	74%
	1 sec	$3,046.27	$10,140.00	$7,093.73	$ 1,521.30	$ 5,572.43	74%
	2 sec	$3,046.27	$10,140.00	$7,093.73	$ 1,521.30	$ 5,572.43	74%
	5 sec	$3,046.27	$10,140.00	$7,093.73	$ 1,521.30	$ 5,572.43	74%
Path cheapest arc & Tabu search	.5 sec	$2,903.04	$10,065.00	$7,161.96	$ 1,452.90	$ 5,709.06	74%
	1 sec	$2,892.44	$10,200.00	$7,307.56	$ 1,379.90	$ 5,927.66	75%
	2 sec	$3,036.60	$10,665.00	$7,628.41	$ 1,286.70	$ 6,341.71	78%
	5 sec	$2,905.27	$10,560.00	$7,654.73	$ 1,281.10	$ 6,373.63	77%
Path cheapest arc & Guided local search	.5 sec	$2,905.13	$10,455.00	$7,549.88	$ 1,304.90	$ 6,244.98	77%
	1 sec	$2,951.54	$10,515.00	$7,563.46	$ 1,827.70	$ 5,735.76	77%
	2 sec	$2,906.83	$10,530.00	$7,623.17	$ 1,377.40	$ 6,245.77	77%
	5 sec	$2,845.24	$10,485.00	$7,639.76	$ 1,335.70	$ 6,304.06	77%
Savings & Simulated annealing	.5 sec	$3,087.77	$ 6,210.00	$3,122.24	$ 3,722.90	$ (600.67)	46%
	1 sec	$3,087.77	$ 6,210.00	$3,122.24	$ 3,722.90	$ (600.67)	46%
	2 sec	$3,087.77	$ 6,210.00	$3,122.24	$ 3,722.90	$ (600.67)	46%
	5 sec	$3,087.77	$ 6,210.00	$3,122.24	$ 3,722.90	$ (600.67)	46%
Savings & Tabu search	.5 sec	$3,055.54	$ 4,140.00	$1,084.46	$ 4,809.40	$(3,724.94)	30%
	1 sec	$2,994.54	$ 6,720.00	$3,725.46	$ 3,409.70	$ 315.76	49%
	2 sec	$3,057.19	$ 7,320.00	$4,262.81	$ 3,278.50	$ 984.31	54%
	5 sec	$2,916.39	$ 7,425.00	$4,508.61	$ 2,935.40	$ 1,573.21	54%
Savings & Guided local search	.5 sec	$3,154.00	$ 7,260.00	$4,106.00	$ 3,161.20	$ 944.80	53%
	1 sec	$2,985.90	$ 8,130.00	$5,144.10	$ 2,726.60	$ 2,417.50	60%
	2 sec	$2,936.30	$ 9,345.00	$6,408.70	$ 2,019.50	$ 4,389.20	69%
	5 sec	$3,002.15	$ 9,795.00	$6,792.85	$ 1,880.50	$ 4,912.35	72%
Christofides & Simulated annealing	.5 sec	$3,093.89	$ 7,830.00	$4,736.11	$ 2,451.40	$ 2,284.71	57%
	1 sec	$3,093.89	$ 7,830.00	$4,736.11	$ 2,451.40	$ 2,284.71	57%
	2 sec	$3,093.89	$ 7,830.00	$4,736.11	$ 2,451.40	$ 2,284.71	57%
	5 sec	$3,093.89	$ 7,830.00	$4,736.11	$ 2,451.40	$ 2,284.71	57%
Christofides & Tabu search	.5 sec	$2,971.73	$ 8,175.00	$5,203.27	$ 2,546.50	$ 2,656.77	60%
	1 sec	$2,905.10	$ 9,525.00	$6,619.90	$ 1,716.90	$ 4,903.00	70%
	2 sec	$2,982.18	$ 8,940.00	$5,957.82	$ 1,995.40	$ 3,962.42	66%
	5 sec	$3,013.42	$ 9,000.00	$5,986.58	$ 2,136.20	$ 3,850.38	66%
Christofides & Guided local search	.5 sec	$3,059.27	$ 9,645.00	$6,585.74	$ 1,709.60	$ 4,876.14	71%
	1 sec	$2,816.84	$ 8,490.00	$5,673.16	$ 2,382.20	$ 3,290.96	62%
	2 sec	$2,925.88	$ 9,855.00	$6,929.12	$ 1,538.40	$ 5,390.72	72%
	5 sec	$3,056.31	$10,005.00	$6,948.69	$ 1,627.40	$ 5,321.29	73%

Table 5.8 shows the various costs for increasing the capacity of vehicle 2 by 20 units. Comparing various adjusted profit for different algorithm combinations, Path cheapest Arc and Tabu search algorithm with a run limit of 5 seconds produced the highest adjusted profit of $6373.63, while serving 77% of the total customers.

5.2.2.3 Increasing capacity of vehicle 3

Table 5.9. Route summary

Algorithm Combination	Algorithm Run time	Distance travelled (miles)					Unsatisfied demand	Dropped visits
		Vehicle 1	Vehicle 2	Vehicle 3	Vehicle 4	Total		
Path cheapest arc & Simulated annealing	.5 sec	421.8	425	422.1	477.9	1746.8	251	33
	1 sec	421.8	425	422.1	477.9	1746.8	251	33
	2 sec	421.8	425	422.1	477.9	1746.8	251	33
	5 sec	421.8	425	422.1	477.9	1746.8	251	33
Path cheapest arc & Tabu search	.5 sec	449.5	475.5	441	446.1	1812.1	279	37
	1 sec	469.5	436.4	469.4	446.3	1821.6	244	33
	2 sec	465.8	443.3	452.1	477.5	1838.7	258	34
	5 sec	464.2	432.7	466.5	461.1	1824.5	240	31
Path cheapest arc & Guided local search	.5 sec	480.8	425	437.5	434.3	1777.6	211	27
	1 sec	463.7	430.2	503.7	424.7	1822.3	174	23
	2 sec	443.2	452.6	508.7	469.6	1874.1	195	26
	5 sec	436.7	423.5	431.8	415.9	1707.9	233	32
Savings & Simulated annealing	.5 sec	499.7	414.7	423.4	475.5	1813.3	651	88
	1 sec	499.7	414.7	423.4	475.5	1813.3	651	88
	2 sec	499.7	414.7	423.4	475.5	1813.3	651	88
	5 sec	499.7	414.7	423.4	475.5	1813.3	651	88
Savings & Tabu search	.5 sec	479.3	492.2	426	468.5	1866	561	76
	1 sec	467.9	460.4	424.3	459	1811.6	627	82
	2 sec	466	507.3	463.4	500.7	1937.4	563	73
	5 sec	442.6	477.6	490.7	445.5	1856.4	468	60
Savings & Guided local search	.5 sec	513.4	469.6	455.5	502.1	1940.6	389	49
	1 sec	446.8	494.9	501.3	438	1881	348	44
	2 sec	451.4	431.8	474.2	496.3	1853.7	356	46
	5 sec	418.8	420.4	490.2	451.6	1781	286	37
Christofides & Simulated annealing	.5 sec	428.8	433.7	496.5	460.2	1819.2	402	55
	1 sec	428.8	433.7	496.5	460.2	1819.2	402	55
	2 sec	428.8	433.7	496.5	460.2	1819.2	402	55
	5 sec	428.8	433.7	496.5	460.2	1819.2	402	55
Christofides & Tabu search	.5 sec	421.4	489.3	487.8	435	1833.5	424	59
	1 sec	537.9	453.6	464.1	426.8	1882.4	424	58
	2 sec	438.2	433	526.9	471.5	1869.6	414	55
	5 sec	492.5	423.4	458.7	439.9	1814.5	374	51
Christofides & Guided local search	.5 sec	426.2	473.8	450.3	452.8	1803.1	307	41
	1 sec	429.4	452.4	466.2	423.7	1771.7	306	42
	2 sec	429	478.2	430.4	444.3	1781.9	288	41
	5 sec	425	448.4	523.6	432.7	1829.7	277	39

The route summary after implementing the decision to increase the capacity of vehicle 3 by 20 units is shown in table 5.9. The combination of Path cheapest arc and Guided local search with a run time of 5 seconds produced the lowest fleet distance of 1707.9 miles. However, this combination dropped 32 customer locations, which equals to 233 unsatisfied quantities.

Table 5.10. Cost summary

Algorithm Combination	Algorithm Run time	Expense	Revenue	Profit	Lost opportunity	Adjusted profit	Service level
Path cheapest arc & Simulated annealing	.5 sec	$2,817.26	$ 9,870.00	$7,052.74	$ 1,749.80	$ 5,302.94	72%
	1 sec	$2,817.26	$ 9,870.00	$7,052.74	$ 1,749.80	$ 5,302.94	72%
	2 sec	$2,817.26	$ 9,870.00	$7,052.74	$ 1,749.80	$ 5,302.94	72%
	5 sec	$2,817.26	$ 9,870.00	$7,052.74	$ 1,749.80	$ 5,302.94	72%
Path cheapest arc & Tabu search	.5 sec	$2,924.10	$ 9,450.00	$6,525.91	$ 1,711.00	$ 4,814.91	69%
	1 sec	$2,945.53	$ 9,975.00	$7,029.47	$ 1,350.90	$ 5,678.57	73%
	2 sec	$2,968.63	$ 9,765.00	$6,796.37	$ 1,553.10	$ 5,243.27	72%
	5 sec	$2,949.08	$10,035.00	$7,085.92	$ 1,512.80	$ 5,573.12	74%
Path cheapest arc & Guided local search	.5 sec	$2,872.49	$10,470.00	$7,597.51	$ 1,369.10	$ 6,228.41	77%
	1 sec	$2,952.33	$11,025.00	$8,072.67	$ 1,149.50	$ 6,923.17	81%
	2 sec	$3,031.88	$10,710.00	$7,678.12	$ 1,314.70	$ 6,363.42	79%
	5 sec	$2,759.91	$10,140.00	$7,380.10	$ 1,492.60	$ 5,887.50	74%
Savings & Simulated annealing	.5 sec	$2,926.80	$ 3,870.00	$ 943.20	$ 4,807.30	$(3,864.10)	28%
	1 sec	$2,926.80	$ 3,870.00	$ 943.20	$ 4,807.30	$(3,864.10)	28%
	2 sec	$2,926.80	$ 3,870.00	$ 943.20	$ 4,807.30	$(3,864.10)	28%
	5 sec	$2,926.80	$ 3,870.00	$ 943.20	$ 4,807.30	$(3,864.10)	28%
Savings & Tabu search	.5 sec	$3,007.32	$ 5,220.00	$2,212.68	$ 4,079.20	$(1,866.52)	38%
	1 sec	$2,921.79	$ 4,230.00	$1,308.21	$ 4,863.60	$(3,555.39)	31%
	2 sec	$3,123.66	$ 5,190.00	$2,066.34	$ 4,377.60	$(2,311.26)	38%
	5 sec	$3,001.15	$ 6,615.00	$3,613.85	$ 3,376.70	$ 237.15	49%
Savings & Guided local search	.5 sec	$3,130.99	$ 7,800.00	$4,669.01	$ 2,905.40	$ 1,763.61	57%
	1 sec	$3,041.53	$ 8,415.00	$5,373.47	$ 2,452.30	$ 2,921.17	62%
	2 sec	$2,994.70	$ 8,295.00	$5,300.30	$ 2,332.50	$ 2,967.80	61%
	5 sec	$2,882.29	$ 9,345.00	$6,462.71	$ 1,920.50	$ 4,542.21	69%
Christofides & Simulated annealing	.5 sec	$2,943.43	$ 7,605.00	$4,661.57	$ 2,684.50	$ 1,977.07	56%
	1 sec	$2,943.43	$ 7,605.00	$4,661.57	$ 2,684.50	$ 1,977.07	56%
	2 sec	$2,943.43	$ 7,605.00	$4,661.57	$ 2,684.50	$ 1,977.07	56%
	5 sec	$2,943.43	$ 7,605.00	$4,661.57	$ 2,684.50	$ 1,977.07	56%
Christofides & Tabu search	.5 sec	$2,963.44	$ 7,275.00	$4,311.56	$ 2,920.20	$ 1,391.36	53%
	1 sec	$3,044.29	$ 7,275.00	$4,230.71	$ 2,698.20	$ 1,532.51	53%
	2 sec	$3,027.78	$ 7,425.00	$4,397.22	$ 2,763.30	$ 1,633.92	54%
	5 sec	$2,934.44	$ 8,025.00	$5,090.56	$ 2,345.50	$ 2,745.06	59%
Christofides & Guided local search	.5 sec	$2,909.96	$ 9,030.00	$6,120.05	$ 1,934.80	$ 4,185.25	66%
	1 sec	$2,864.40	$ 9,045.00	$6,180.60	$ 2,066.70	$ 4,113.90	66%
	2 sec	$2,873.74	$ 9,315.00	$6,441.26	$ 1,641.30	$ 4,799.96	68%
	5 sec	$2,964.28	$ 9,480.00	$6,515.72	$ 1,683.60	$ 4,832.12	70%

The cost summary is shown in table 5.10, where the combination of Path cheapest arc and Guided local search with a run time of 1 second produced the highest adjusted profit of $6923.17. This combination served 81% of the customers while the vehicle fleet traveled a total distance of 1822.3 miles.

5.2.2.4 Increasing capacity of vehicle 4

Table 5.11. Route summary

Algorithm Combination	Algorithm Run time	Distance travelled (miles)					Unsatisfied demand	Dropped visits
		Vehicle 1	Vehicle 2	Vehicle 3	Vehicle 4	Total		
Path cheapest arc & Simulated annealing	.5 sec	494.2	425.5	447.6	437.7	1805	193	25
	1 sec	494.2	425.5	447.6	437.7	1805	193	25
	2 sec	494.2	425.5	447.6	437.7	1805	193	25
	5 sec	494.2	425.5	447.6	437.7	1805	193	25
Path cheapest arc & Tabu search	.5 sec	414.8	446	436.6	514.9	1812.3	227	30
	1 sec	429.8	457.2	440.5	437.3	1764.8	259	35
	2 sec	457.6	420	428.5	473.1	1779.2	233	32
	5 sec	451.3	446.3	457.5	462.1	1817.2	176	24
Path cheapest arc & Guided local search	.5 sec	455.5	440.1	463.6	481.7	1840.9	218	29
	1 sec	432.3	431.1	507.9	439.4	1810.7	186	25
	2 sec	431.4	433	484.4	452.7	1801.5	215	29
	5 sec	416.4	418.3	421.6	507.4	1763.7	226	29
Savings & Simulated annealing	.5 sec	461.2	489.2	485.9	487.1	1923.4	665	91
	1 sec	461.2	489.2	485.9	487.1	1923.4	665	91
	2 sec	461.2	489.2	485.9	487.1	1923.4	665	91
	5 sec	461.2	489.2	485.9	487.1	1923.4	665	91
Savings & Tabu search	.5 sec	511.7	416.5	441.7	487.7	1857.6	548	71
	1 sec	477.1	510.3	488.9	478.7	1955	548	73
	2 sec	456	451.7	487.4	453	1848.1	349	44
	5 sec	478.5	462.7	443.2	451.2	1835.6	338	42
Savings & Guided local search	.5 sec	427.4	475.3	481.1	462.4	1846.2	406	52
	1 sec	427	457	484.5	433.8	1802.3	373	46
	2 sec	434.8	454.7	414.3	457	1760.8	359	46
	5 sec	425.7	465.4	418.5	467.4	1777	260	34
Christofides & Simulated annealing	.5 sec	452.4	421.8	427.2	475.5	1776.9	418	59
	1 sec	452.4	421.8	427.2	475.5	1776.9	418	59
	2 sec	452.4	421.8	427.2	475.5	1776.9	418	59
	5 sec	452.4	421.8	427.2	475.5	1776.9	418	59
Christofides & Tabu search	.5 sec	443.3	485.4	462	451.5	1842.2	413	56
	1 sec	421.6	502.7	470.8	496.1	1891.2	348	50
	2 sec	421.6	502.7	470.8	496.1	1891.2	348	50
	5 sec	458.9	414.1	474.3	477	1824.3	309	43
Christofides & Guided local search	.5 sec	421.5	446.1	477.8	424.8	1770.2	329	44
	1 sec	475.3	440.2	455.3	431.3	1802.1	298	40
	2 sec	470	465.3	455.1	463.7	1854.1	320	44
	5 sec	482.6	449.8	461.4	452.8	1846.6	269	37

The combination of Path Cheapest Arc and Guided Local Search with a run time of 5 seconds produced the shortest total fleet distance of 1763.7 miles, as seen in table 26. This combination served 119 customer requests while dropping 29 customer locations.Below table shows the various costs associated with this decision.

Table 5.12. Cost summary

Algorithm Combination	Algorithm Run time	Expense	Revenue	Profit	Lost opportunity	Adjusted profit	Service level
Path cheapest arc & Simulated annealing	.5 sec	$2,916.86	$10,740.00	$7,823.14	$ 1,233.00	$ 6,590.14	79%
	1 sec	$2,916.86	$10,740.00	$7,823.14	$ 1,233.00	$ 6,590.14	79%
	2 sec	$2,916.86	$10,740.00	$7,823.14	$ 1,233.00	$ 6,590.14	79%
	5 sec	$2,916.86	$10,740.00	$7,823.14	$ 1,233.00	$ 6,590.14	79%
Path cheapest arc & Tabu search	.5 sec	$2,928.67	$10,230.00	$7,301.33	$ 1,664.50	$ 5,636.83	75%
	1 sec	$2,848.79	$ 9,750.00	$6,901.21	$ 1,754.30	$ 5,146.91	72%
	2 sec	$2,875.96	$10,140.00	$7,264.04	$ 1,460.60	$ 5,803.44	74%
	5 sec	$2,935.57	$10,995.00	$8,059.43	$ 1,214.00	$ 6,845.43	81%
Path cheapest arc & Guided local search	.5 sec	$2,975.09	$10,365.00	$7,389.91	$ 1,430.00	$ 5,959.91	76%
	1 sec	$2,925.74	$10,845.00	$7,919.26	$ 1,111.20	$ 6,808.06	80%
	2 sec	$2,910.86	$10,410.00	$7,499.14	$ 1,534.30	$ 5,964.84	76%
	5 sec	$2,851.52	$10,245.00	$7,393.49	$ 1,466.80	$ 5,926.69	75%
Savings & Simulated annealing	.5 sec	$3,105.75	$ 3,660.00	$ 554.25	$ 5,038.50	$(4,484.25)	27%
	1 sec	$3,105.75	$ 3,660.00	$ 554.25	$ 5,038.50	$(4,484.25)	27%
	2 sec	$3,105.75	$ 3,660.00	$ 554.25	$ 5,038.50	$(4,484.25)	27%
	5 sec	$3,105.75	$ 3,660.00	$ 554.25	$ 5,038.50	$(4,484.25)	27%
Savings & Tabu search	.5 sec	$3,004.32	$ 5,415.00	$2,410.68	$ 4,053.90	$(1,643.22)	40%
	1 sec	$3,155.40	$ 5,415.00	$2,259.60	$ 3,993.30	$(1,733.70)	40%
	2 sec	$2,985.33	$ 8,400.00	$5,414.67	$ 2,469.40	$ 2,945.27	62%
	5 sec	$2,964.04	$ 8,565.00	$5,600.96	$ 2,530.40	$ 3,070.56	63%
Savings & Guided local search	.5 sec	$2,980.53	$ 7,545.00	$4,564.47	$ 3,094.50	$ 1,469.97	55%
	1 sec	$2,909.87	$ 8,040.00	$5,130.14	$ 2,741.90	$ 2,388.24	59%
	2 sec	$2,842.87	$ 8,250.00	$5,407.13	$ 2,577.70	$ 2,829.43	61%
	5 sec	$2,868.63	$ 9,735.00	$6,866.37	$ 1,825.00	$ 5,041.37	71%
Christofides & Simulated annealing	.5 sec	$2,872.11	$ 7,365.00	$4,492.89	$ 2,825.90	$ 1,666.99	54%
	1 sec	$2,872.11	$ 7,365.00	$4,492.89	$ 2,825.90	$ 1,666.99	54%
	2 sec	$2,872.11	$ 7,365.00	$4,492.89	$ 2,825.90	$ 1,666.99	54%
	5 sec	$2,872.11	$ 7,365.00	$4,492.89	$ 2,825.90	$ 1,666.99	54%
Christofides & Tabu search	.5 sec	$2,972.98	$ 7,440.00	$4,467.02	$ 3,051.90	$ 1,415.12	55%
	1 sec	$3,052.36	$ 8,415.00	$5,362.64	$ 2,207.30	$ 3,155.34	62%
	2 sec	$3,052.36	$ 8,415.00	$5,362.64	$ 2,207.30	$ 3,155.34	62%
	5 sec	$2,950.02	$ 9,000.00	$6,049.98	$ 2,061.10	$ 3,988.88	66%
Christofides & Guided local search	.5 sec	$2,858.23	$ 8,700.00	$5,841.77	$ 2,342.50	$ 3,499.27	64%
	1 sec	$2,910.83	$ 9,165.00	$6,254.17	$ 1,891.00	$ 4,363.17	67%
	2 sec	$2,994.23	$ 8,835.00	$5,840.77	$ 2,135.90	$ 3,704.87	65%
	5 sec	$2,983.03	$ 9,600.00	$6,616.97	$ 1,677.80	$ 4,939.17	70%

The combination of Path Cheapest Arc and Tabu search with 5 second run time produced the highest adjusted profit value of $6845.43 while serving 81% of the customers and producing a total fleet distance of 1817.2 miles. The details of the associated costs are shown in table 5.12. The next section discusses the results while adding an extra vehicle to the vehicle fleet.

5.2.3 Decision 2 - Adding an extra vehicle at each depot

This section discusses the two scenarios that arise when implementing the decision to add an extra vehicle at each depot. For this study, an additional vehicle with capacity 30 is added at each depot.

5.2.3.1 Adding an extra vehicle at depot 1

Table 5.13. Route summary

Algorithm Combination	Algorithm Run time	Distance travelled (miles)						Unsatisfied demand	Dropped visits
		Vehicle 1	Vehicle 2	Vehicle 3	Vehicle 4	Vehicle 5	Total		
Path cheapest arc & Simulated annealing	.5 sec	444.2	434.5	470.6	468.4	529.5	2347.2	124	16
	1 sec	444.2	434.5	470.6	468.4	529.5	2347.2	124	16
	2 sec	444.2	434.5	470.6	468.4	529.5	2347.2	124	16
	5 sec	444.2	434.5	470.6	468.4	529.5	2347.2	124	16
Path cheapest arc & Tabu search	.5 sec	446.5	424.5	437.8	489.9	483.3	2282	154	20
	1 sec	454.7	428.8	446.4	490.7	509.9	2330.5	185	22
	2 sec	468.2	431.9	485	482.7	447.1	2314.9	165	21
	5 sec	492.4	480.8	427.8	454.9	462.4	2318.3	125	16
Path cheapest arc & Guided local search	.5 sec	504.2	441.5	441.4	425.5	460.7	2273.3	232	30
	1 sec	422.6	488.4	414.7	430.8	475.5	2232	152	20
	2 sec	426.6	469.2	458.3	433.8	420.9	2208.8	137	19
	5 sec	471.4	472.3	492.3	442.3	441.7	2320	79	10
Savings & Simulated annealing	.5 sec	418.2	448.2	474.7	520.3	495.9	2357.3	584	77
	1 sec	418.2	448.2	474.7	520.3	495.9	2357.3	584	77
	2 sec	418.2	448.2	474.7	520.3	495.9	2357.3	584	77
	5 sec	418.2	448.2	474.7	520.3	495.9	2357.3	584	77
Savings & Tabu search	.5 sec	467	449.4	453.2	455.7	477.9	2303.2	458	58
	1 sec	511.6	470.9	422.5	425.5	468.7	2299.2	393	50
	2 sec	451.1	419.7	431.4	463.6	527.7	2293.5	362	44
	5 sec	516.8	415.9	416.3	428.9	466.8	2244.7	301	36
Savings & Guided local search	.5 sec	462.7	449.7	453	420.5	463.3	2249.2	286	36
	1 sec	425.6	442.5	493.3	447.1	444.5	2253	279	34
	2 sec	421.4	439.7	421.7	497.8	460.2	2240.8	199	25
	5 sec	452.6	434	454.4	436.6	430.6	2208.2	239	30
Christofides & Simulated annealing	.5 sec	469.8	456.8	488	451.8	459.6	2326	322	44
	1 sec	469.8	456.8	488	451.8	459.6	2326	322	44
	2 sec	469.8	456.8	488	451.8	459.6	2326	322	44
	5 sec	469.8	456.8	488	451.8	459.6	2326	322	44
Christofides & Tabu search	.5 sec	473.3	444	465.1	455.2	443.4	2281	355	48
	1 sec	466.7	447.6	493.5	415.3	504.3	2327.4	303	38
	2 sec	498.9	470.2	415.5	442.5	483	2310.1	313	40
	5 sec	463.3	482.9	517	449.1	441.2	2353.5	215	26
Christofides & Guided local search	.5 sec	469.6	466.3	482.2	464.3	492.8	2375.2	202	27
	1 sec	478.6	444.5	434.4	449.8	470.2	2277.5	194	28
	2 sec	457.9	458.9	477.4	459.7	439.9	2293.8	206	29
	5 sec	422.3	508.5	438.1	428.1	426.9	2223.9	203	27

Table 5.14. Cost summary

Algorithm Combination	Algorithm Run time	Expense	Revenue	Profit	Lost opportunity	Adjusted profit	Service level
Path cheapest arc & Simulated annealing	.5 sec	$3,753.70	$11,775.00	$8,021.30	$ 707.90	$ 7,313.40	86%
	1 sec	$3,753.70	$11,775.00	$8,021.30	$ 707.90	$ 7,313.40	86%
	2 sec	$3,753.70	$11,775.00	$8,021.30	$ 707.90	$ 7,313.40	86%
	5 sec	$3,753.70	$11,775.00	$8,021.30	$ 707.90	$ 7,313.40	86%
Path cheapest arc & Tabu search	.5 sec	$3,646.51	$11,325.00	$7,678.49	$ 916.20	$ 6,762.29	83%
	1 sec	$3,725.16	$10,860.00	$7,134.85	$ 1,161.00	$ 5,973.85	80%
	2 sec	$3,700.12	$11,160.00	$7,459.88	$ 1,080.90	$ 6,378.98	82%
	5 sec	$3,703.97	$11,760.00	$8,056.03	$ 731.10	$ 7,324.93	86%
Path cheapest arc & Guided local search	.5 sec	$3,636.12	$10,155.00	$6,518.88	$ 1,563.90	$ 4,954.98	74%
	1 sec	$3,564.62	$11,355.00	$7,790.38	$ 1,014.70	$ 6,775.68	83%
	2 sec	$3,528.10	$11,580.00	$8,051.90	$ 810.50	$ 7,241.40	85%
	5 sec	$3,708.43	$12,450.00	$8,741.57	$ 523.60	$ 8,217.97	91%
Savings & Simulated annealing	.5 sec	$3,764.92	$ 4,875.00	$1,110.08	$ 4,400.20	$(3,290.12)	36%
	1 sec	$3,764.92	$ 4,875.00	$1,110.08	$ 4,400.20	$(3,290.12)	36%
	2 sec	$3,764.92	$ 4,875.00	$1,110.08	$ 4,400.20	$(3,290.12)	36%
	5 sec	$3,764.92	$ 4,875.00	$1,110.08	$ 4,400.20	$(3,290.12)	36%
Savings & Tabu search	.5 sec	$3,681.81	$ 6,765.00	$3,083.19	$ 3,475.90	$ (392.71)	50%
	1 sec	$3,675.98	$ 7,740.00	$4,064.02	$ 2,837.80	$ 1,226.22	57%
	2 sec	$3,667.74	$ 8,205.00	$4,537.26	$ 2,853.60	$ 1,683.66	60%
	5 sec	$3,591.28	$ 9,120.00	$5,528.72	$ 2,263.10	$ 3,265.62	67%
Savings & Guided local search	.5 sec	$3,596.58	$ 9,345.00	$5,748.42	$ 2,085.20	$ 3,663.22	69%
	1 sec	$3,601.22	$ 9,450.00	$5,848.78	$ 2,022.90	$ 3,825.88	69%
	2 sec	$3,577.50	$10,650.00	$7,072.50	$ 1,355.00	$ 5,717.50	78%
	5 sec	$3,529.72	$10,050.00	$6,520.28	$ 1,814.40	$ 4,705.88	74%
Christofides & Simulated annealing	.5 sec	$3,718.69	$ 8,805.00	$5,086.31	$ 2,127.00	$ 2,959.31	65%
	1 sec	$3,718.69	$ 8,805.00	$5,086.31	$ 2,127.00	$ 2,959.31	65%
	2 sec	$3,718.69	$ 8,805.00	$5,086.31	$ 2,127.00	$ 2,959.31	65%
	5 sec	$3,718.69	$ 8,805.00	$5,086.31	$ 2,127.00	$ 2,959.31	65%
Christofides & Tabu search	.5 sec	$3,646.09	$ 8,310.00	$4,663.91	$ 2,331.50	$ 2,332.41	61%
	1 sec	$3,724.63	$ 9,090.00	$5,365.37	$ 2,102.10	$ 3,263.27	67%
	2 sec	$3,692.45	$ 8,940.00	$5,247.55	$ 2,232.40	$ 3,015.15	66%
	5 sec	$3,761.65	$10,410.00	$6,648.36	$ 1,662.50	$ 4,985.86	76%
Christofides & Guided local search	.5 sec	$3,797.13	$10,605.00	$6,807.87	$ 1,434.10	$ 5,373.77	78%
	1 sec	$3,640.78	$10,725.00	$7,084.22	$ 1,113.20	$ 5,971.02	79%
	2 sec	$3,665.41	$10,545.00	$6,879.59	$ 1,215.50	$ 5,664.09	77%
	5 sec	$3,550.03	$10,590.00	$7,039.97	$ 1,305.00	$ 5,734.97	78%

The combination of Savings and Guided local search produced the lowest fleet distance of 2208.2 miles while serving 74% of the customer requests. However, the combination of Path cheapest Arc and Guided local search produced the highest adjusted profit of $8217.97 while serving 91 % of the customer request, which is the highest service level among all the decisions.

5.2.3.2 Adding an extra vehicle at depot 2

Table 5.15. Route summary

Algorithm Combination	Algorithm Run time	Distance travelled (miles)						Unsatisfied demand	Dropped visits
		Vehicle 1	Vehicle 2	Vehicle 3	Vehicle 4	Vehicle 5	Total		
Path cheapest arc & Simulated annealing	.5 sec	473.3	456	470.7	436.3	423.2	2259.5	113	16
	1 sec	473.3	456	470.7	436.3	423.2	2259.5	113	16
	2 sec	473.3	456	470.7	436.3	423.2	2259.5	113	16
	5 sec	473.3	456	470.7	436.3	423.2	2259.5	113	16
Path cheapest arc & Tabu search	.5 sec	476.9	418	439.4	433.4	429.9	2197.6	118	17
	1 sec	451.5	468.5	431	463	490	2304	128	17
	2 sec	423.7	428.2	425.9	449.7	447.2	2174.7	265	34
	5 sec	419	470.8	438.6	434	452.3	2214.7	199	27
Path cheapest arc & Guided local search	.5 sec	427	427.5	427	454.3	448.7	2184.5	152	19
	1 sec	418	426.7	505.7	448.4	492.7	2291.5	108	15
	2 sec	431.9	445.1	423.4	467.2	440.9	2208.5	148	21
	5 sec	476.7	428.8	428.1	415.2	484.3	2233.1	88	12
Savings & Simulated annealing	.5 sec	465.6	422.7	496.7	431.1	528.5	2344.6	461	59
	1 sec	465.6	422.7	496.7	431.1	528.5	2344.6	461	59
	2 sec	465.6	422.7	496.7	431.1	528.5	2344.6	461	59
	5 sec	465.6	422.7	496.7	431.1	528.5	2344.6	461	59
Savings & Tabu search	.5 sec	509.8	455.7	487.3	465	469.1	2386.9	442	56
	1 sec	457.7	448.4	464.3	457.1	445.6	2273.1	407	51
	2 sec	448.2	445	457.6	456	479.6	2286.4	310	39
	5 sec	415.9	467.9	470.6	437.5	466.7	2258.6	266	34
Savings & Guided local search	.5 sec	429.7	433.8	496	477	427.7	2264.2	279	35
	1 sec	441.6	438.7	465.8	446.1	463.7	2255.9	258	33
	2 sec	489	444.1	464.8	440.5	440.3	2278.7	268	32
	5 sec	478.9	443.4	440.8	472.3	447.7	2283.1	188	25
Christofides & Simulated annealing	.5 sec	430.2	500.6	483.4	455.4	474.3	2343.9	289	39
	1 sec	430.2	500.6	483.4	455.4	474.3	2343.9	289	39
	2 sec	430.2	500.6	483.4	455.4	474.3	2343.9	289	39
	5 sec	430.2	500.6	483.4	455.4	474.3	2343.9	289	39
Christofides & Tabu search	.5 sec	501	478.5	434.1	455.2	418.9	2287.7	318	43
	1 sec	510	438.2	449.8	415.4	474.4	2287.8	211	29
	2 sec	436.9	473	482.9	485.1	459.1	2337	218	31
	5 sec	419.6	466	491.6	427.8	421.6	2226.6	239	33
Christofides & Guided local search	.5 sec	472.6	457.4	443.5	444.5	455.4	2273.4	206	27
	1 sec	431.5	497.6	445.5	454.7	459.6	2288.9	225	33
	2 sec	468.1	430.1	463.6	432.8	490.5	2285.1	185	26
	5 sec	425.8	430.1	429	469.3	484.8	2239	155	21

Table 5.15 tabulates the route summary for implementing the decision to add an extra vehicle with capacity 30 at depot 2. The combination of Path cheapest arc and Tabu search for a run time of 2 seconds produces the lowest fleet distance of 2174.7 miles. However, this algorithm combination dropped 34 customer requests, which equal to 265 unsatisfied quantities. The below table shows the various costs associated with adding an extra vehicle at depot 2.

Table 5.16. Cost summary

Algorithm Combination	Algorithm Run time	Expense	Revenue	Profit	Lost opportunity	Adjusted profit	Service level
Path cheapest arc & Simulated annealing	.5 sec	$3,611.60	$11,940.00	$8,328.40	$ 629.10	$ 7,699.30	88%
	1 sec	$3,611.60	$11,940.00	$8,328.40	$ 629.10	$ 7,699.30	88%
	2 sec	$3,611.60	$11,940.00	$8,328.40	$ 629.10	$ 7,699.30	88%
	5 sec	$3,611.60	$11,940.00	$8,328.40	$ 629.10	$ 7,699.30	88%
Path cheapest arc & Tabu search	.5 sec	$3,513.98	$11,865.00	$8,351.02	$ 732.10	$ 7,618.92	87%
	1 sec	$3,681.00	$11,715.00	$8,034.00	$ 746.30	$ 7,287.70	86%
	2 sec	$3,474.53	$ 9,660.00	$6,185.47	$ 1,792.90	$ 4,392.57	71%
	5 sec	$3,537.58	$10,650.00	$7,112.42	$ 1,274.10	$ 5,838.32	78%
Path cheapest arc & Guided local search	.5 sec	$3,490.19	$11,355.00	$7,864.81	$ 1,080.80	$ 6,784.01	83%
	1 sec	$3,665.14	$12,015.00	$8,349.86	$ 585.00	$ 7,764.86	88%
	2 sec	$3,526.87	$11,415.00	$7,888.13	$ 863.70	$ 7,024.43	84%
	5 sec	$3,572.43	$12,315.00	$8,742.57	$ 539.10	$ 8,203.47	90%
Savings & Simulated annealing	.5 sec	$3,753.39	$ 6,720.00	$2,966.61	$ 3,413.40	$ (446.79)	49%
	1 sec	$3,753.39	$ 6,720.00	$2,966.61	$ 3,413.40	$ (446.79)	49%
	2 sec	$3,753.39	$ 6,720.00	$2,966.61	$ 3,413.40	$ (446.79)	49%
	5 sec	$3,753.39	$ 6,720.00	$2,966.61	$ 3,413.40	$ (446.79)	49%
Savings & Tabu search	.5 sec	$3,816.99	$ 7,005.00	$3,188.01	$ 3,309.60	$ (121.59)	51%
	1 sec	$3,632.71	$ 7,530.00	$3,897.29	$ 2,892.50	$ 1,004.79	55%
	2 sec	$3,654.75	$ 8,985.00	$5,330.25	$ 2,128.70	$ 3,201.55	66%
	5 sec	$3,609.09	$ 9,645.00	$6,035.91	$ 1,891.60	$ 4,144.31	71%
Savings & Guided local search	.5 sec	$3,617.78	$ 9,450.00	$5,832.22	$ 1,977.40	$ 3,854.82	69%
	1 sec	$3,606.33	$ 9,765.00	$6,158.67	$ 1,882.00	$ 4,276.67	72%
	2 sec	$3,643.51	$ 9,615.00	$5,971.49	$ 2,195.70	$ 3,775.79	71%
	5 sec	$3,648.20	$10,815.00	$7,166.80	$ 1,364.10	$ 5,802.70	79%
Christofides & Simulated annealing	.5 sec	$3,744.08	$ 9,300.00	$5,555.92	$ 1,883.40	$ 3,672.52	68%
	1 sec	$3,744.08	$ 9,300.00	$5,555.92	$ 1,883.40	$ 3,672.52	68%
	2 sec	$3,744.08	$ 9,300.00	$5,555.92	$ 1,883.40	$ 3,672.52	68%
	5 sec	$3,744.08	$ 9,300.00	$5,555.92	$ 1,883.40	$ 3,672.52	68%
Christofides & Tabu search	.5 sec	$3,654.26	$ 8,865.00	$5,210.75	$ 2,080.20	$ 3,130.55	65%
	1 sec	$3,660.83	$10,470.00	$6,809.17	$ 1,363.90	$ 5,445.27	77%
	2 sec	$3,732.66	$10,365.00	$6,632.34	$ 1,473.90	$ 5,158.44	76%
	5 sec	$3,557.85	$10,050.00	$6,492.15	$ 1,498.10	$ 4,994.05	74%
Christofides & Guided local search	.5 sec	$3,633.49	$10,545.00	$6,911.51	$ 1,438.70	$ 5,472.81	77%
	1 sec	$3,654.72	$10,260.00	$6,605.28	$ 1,498.80	$ 5,106.48	75%
	2 sec	$3,655.68	$10,860.00	$7,204.32	$ 1,046.40	$ 6,157.92	80%
	5 sec	$3,577.62	$11,310.00	$7,732.38	$ 959.70	$ 6,772.68	83%

From table 5.16, the combination of Path cheapest arc and Guided local search with a run time of 5 seconds produced the highest adjusted profit of $8203.47 while serving 90% of the customer requests. This combination resulted in a total fleet distance of 2233.1 miles. The following section summarises the results for implementing the two decisions.

5.2.4 Result analysis

The second case study studies the performance of the 2-stage decision algorithm for a large dataset when compared to the first case study. Moreover, the second case study explores the various potential decisions from a financial standpoint that could improve the profit as well as the service level of the business. This section analyses the decisions based on both distance traveled by the vehicle fleet and the adjusted profit of the business. Table 5.17 tabulates the best results based on vehicle fleet distance, whereas table 5.18 tabulates the best results based on adjusted profit.

Table 5.17. Best results based on distance

Decesion	Algorithm Combination	Algorithm Run time	Distance travelled (miles)						Service Level
			Vehicle 1	Vehicle 2	Vehicle 3	Vehicle 4	Vehicle 5	Total	
Increasing capacity of V1	Savings & Guided local search	5 sec	450.4	429.9	421.1	439.3	NA	1740.7	62%
Increasing capacity of V2	Christofides & Guided local search	1 sec	428.8	450.8	420.5	444.3	NA	1744.4	62%
Increasing capacity of V3	Path cheapest arc & Guided local search	5 sec	436.7	423.5	431.8	415.9	NA	1707.9	74%
Increasing capacity of V4	Path cheapest arc & Guided local search	5 sec	416.4	418.3	421.6	507.4	NA	1763.7	75%
Adding extra vehicle at depot 1	Savings & Guided local search	5 sec	452.6	434	454.4	436.6	430.6	2208.2	74%
Adding extra vehicle at depot 2	Path cheapest arc & Tabu search	2 sec	423.7	428.2	425.9	449.7	447.2	2174.7	71%

Comparing the total fleet distances between different decisions, the decision to increase the capacity of vehicle by 20 units produces the lowest total fleet distance and simultaneously produces the highest service level of 75%. However, these decisions were implemented to select the most cost-effective method to improve the service level when compared to the initial case without any decisions. Since table 5.17 does not take into account the costs related to the decisions, table 5.18 tabulates the best results among various decisions based on adjusted profit. The adjusted profit takes into account both the distance traveled by the vehicle fleet as well as the service level, which helps to draw a better conclusion on the performance of the 2-stage decision algorithm.

Table 5.18. Best results based on adjusted profits

Decision	**Algorithm Combination**	**Algorithm Run time**	**Expense**	**Revenue**	**Profit**	**Lost opportunity**	**Adjusted profit**	**Service level**
Increasing capacity of V1	Path cheapest arc & Guided local search	0.5 sec	$2,983.46	$11,070.00	$8,086.54	$ 1,081.30	$7,005.24	81%
Increasing capacity of V2	Path cheapest arc & Tabu search	5 sec	$2,905.27	$10,560.00	$7,654.73	$ 1,281.10	$6,373.63	77%
Increasing capacity of V3	Path cheapest arc & Guided local search	1 sec	$2,952.33	$11,025.00	$8,072.67	$ 1,149.50	$6,923.17	81%
Increasing capacity of V4	Path cheapest arc & Tabu search	5 sec	$2,935.57	$10,995.00	$8,059.43	$ 1,214.00	$6,845.43	81%
Adding extra vehicle at depot 1	Path cheapest arc & Guided local search	5 sec	$3,708.43	$12,450.00	$8,741.57	$ 523.00	$8,217.97	91%
Adding extra vehicle at depot 2	Path cheapest arc & Guided local search	5 sec	$3,572.43	$12,315.00	$8,742.57	$ 539.10	$8,203.47	90%

Based on the adjusted profit from table 5.18, the highest adjusted profit is produced by the decision to add an extra vehicle at depot 1. The combination of Path cheapest arc and Guided local search with a run time of 5 seconds produced an adjusted profit of $8,217 with a service level of 91%. This is a 2.5% increase over the adjusted profit when compared to the initial scenario without any decisions. Moreover, the service level improved by 5% over the 86% produced by the initial scenario without any decisions. Figure 24 provides a pictorial representation of the service level and adjusted profit based on table 5.18.

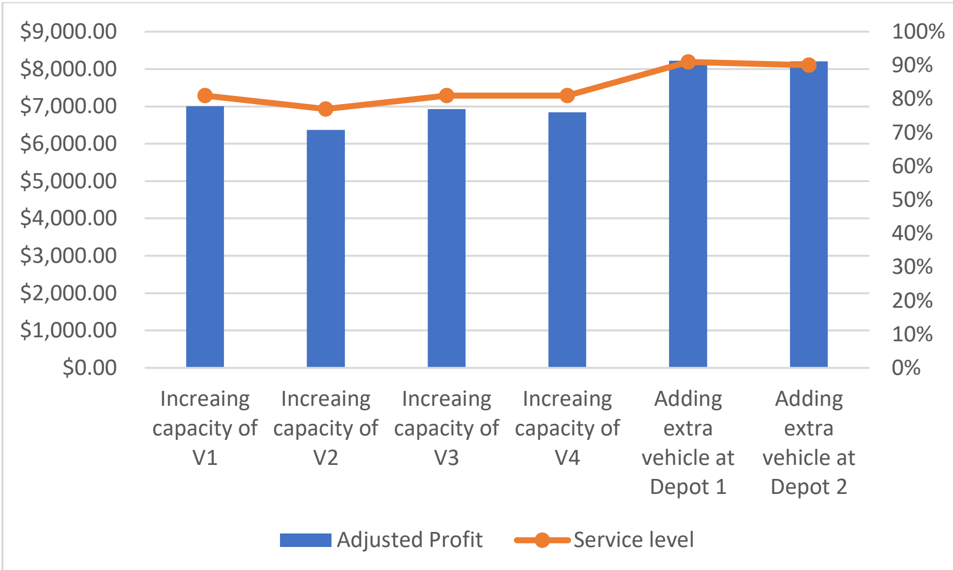

Figure 5.11. Adjusted profit and Service level based on table 33

Comparing tables 5.17 and 5.18, the algorithm combination that produces the highest adjusted profit are different on both the tables. The algorithms on table 5.17 produce the lowest fleet distances at the expense of service level. The maximum service level seen on table 32 is 75% for the decision to increase the capacity of vehicle 4 by 20 units. Nevertheless, the same decision on table 5.18 produces a better service level of 81% as well as a 15% improved adjusted profit using the algorithm combination of Path Cheapest Arc and Tabu search. The reason for the lower adjusted profit for the algorithm on table 5.17 is because, it results in a higher lost opportunity cost and lower revenue due to lower service level. The lost opportunity cost for the combination of Path Cheapest Arc and Guided local search with a 5 second run time is $1446.8, and the lost opportunity cost for the algorithm combination of Path Cheapest Arc and Tabu search with a 5 second run time is $1214. Similarly, the revenue for the comnination of Path Cheapest Arc and Guided local search with a 5 second run time is $10,245, and the revenue for the algorithm combination of Path Cheapest Arc and Tabu search with a 5 second run time is $10,995.

5.3 Conclusion

The Dynamic variant of the MDVRP is a crucial factor in logistics due to its practical and economic significance. In the logistics sector, proper planning of the vehicle fleet is indispensable as it accords in reducing costs and efficient utilization of resources. The work proposed an algorithm to solve an MDVRP that enables to incorporate new customer service requests after all the routes have been planned. A new 2-stage algorithm that could make use of various heuristics and metaheuristics was developed and described.

The proposed 2-stage algorithm can handle predetermined and real-time customer service requests using two stages, as seen in the first case study. The first stage creates an initial route to serve all available customers using anyone among the three heuristics algorithms. Then the second stage uses the routes created by the first stage to try and improve the solution to obtain a better result using one among the three metaheuristics. Since the algorithm deals with dynamic scenarios, computational run time is also taken into consideration. The algorithm is re-run each time a new customer request arrives, or an existing request gets canceled. The proposed methodology is implemented in a case study that depicts a courier collection business to plan the routes for the business's vehicle fleet to handle both predetermined and real-time customer requests.

The second case study investigates various potential decisions to be made regarding the vehicle fleet to serve more customer locations in a cost-effective manner when there is an increase in demand. A cost analysis that takes into account takes into account the revenue, expenses, profit, and the lost opportunity cost is done using the proposed model to see whether to increase the number of vehicles or to increase the individual vehicle capacity to serve all the customers with the minimum cost.

Experimental results tabulated in the previous section using different combinations of heuristics and metaheuristics that shows how the 2-stage algorithm could be used to compare and select the best result to handle the real-time requests.

5.3 Future Research

As part of future research, different financial aspects of D-MDVRP in the areas of vehicle fleet management and depot locations are to be studied. Currently, two depots are serving various customer locations. Other feasible locations can be selected based on a comparative study that takes into account routing and location costs using the python model. Finally, it would be interesting to investigate other algorithms that can yield better results and can be incorporated into the 2-stage algorithm to be used in a dynamic context.

REFERENCES

REFERENCES

AbdElAziz, M. M., A. El-Ghareeb, H., & Ksasy, M. S. M. (2014). Hybrid Heuristic Algorithm for solving Capacitated Vehicle Routing problem. *INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY*. https://doi.org/10.24297/ijct.v12i9.2824

Attanasio, A., Cordeau, J. F., Ghiani, G., & Laporte, G. (2004). Parallel Tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. *Parallel Computing*. https://doi.org/10.1016/j.parco.2003.12.001

Barceló, J., Grzybowska, H., & Pardo, S. (2007). Vehicle Routing and scheduling models, simulation and City Logistics. *Operations Research/ Computer Science Interfaces Series*. https://doi.org/10.1007/978-0-387-71722-7_8

Barreto, L., Amaral, A., & Pereira, T. (2017). Industry 4.0 implications in logistics: an overview. *Procedia Manufacturing*. https://doi.org/10.1016/j.promfg.2017.09.045

Bertsimas, D. J., & van Ryzin, G. (1991). A Stochastic and Dynamic Vehicle Routing Problem in the Euclidean Plane. *Operations Research*. https://doi.org/10.1287/opre.39.4.601

Cassettari, L., Demartini, M., Mosca, R., Revetria, R., & Tonelli, F. (2018). A multi-stage algorithm for a capacitated vehicle routing problem with time constraints. *Algorithms*. https://doi.org/10.3390/a11050069

Christofides, N. (1976). THE VEHBCIE ROUTIWG PROBLEEM (*). *Recherche Opérationnelle*.

Clarke, G., & Wright, J. W. (1964). Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operations Research*. https://doi.org/10.1287/opre.12.4.568

Dantzig, G. B., & Ramser, J. H. (1959). The Truck Dispatching Problem. *Management Science*. https://doi.org/10.1287/mnsc.6.1.80

Ghiani, G., Guerriero, F., Laporte, G., & Musmanno, R. (2003). Real-time vehicle routing: Solution concepts, algorithms and parallel computing strategies. *European Journal of Operational Research*. https://doi.org/10.1016/S0377-2217(02)00915-3

Giosa, I. D., Tansini, I. L., & Viera, I. O. (2002). New assignment algorithms for the multi-depot vehicle routing problem. *Journal of the Operational Research Society*. https://doi.org/10.1057/palgrave.jors.2601426

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*. https://doi.org/10.1016/0305-0548(86)90048-1

Golden, B. L., Magnanti, T. L., & Nguyen, H. Q. (1977). Implementing vehicle routing algorithms. *Networks*. https://doi.org/10.1002/net.3230070203

Haghani, A., & Jung, S. (2005). A dynamic vehicle routing problem with time-dependent travel times. *Computers and Operations Research*. https://doi.org/10.1016/j.cor.2004.04.013

Jeon, G., Leep, H. R., & Shim, J. Y. (2007). A vehicle routing problem solved by using a hybrid genetic algorithm. *Computers and Industrial Engineering*. https://doi.org/10.1016/j.cie.2007.06.031

Kilby, P., Prosser, P., & Shaw, P. (1999). Guided Local Search for the Vehicle Routing Problem with Time Windows. In *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*. https://doi.org/10.1007/978-1-4615-5775-3_32

Kuo, Y., & Wang, C.-C. (2014). Using Insertion Heuristic to Solve Dynamic Multi-Depot Vehicle Routing Problem. *Journal of Algorithms and Optimization*.

Laporte, G., Nobert Yves, & Desrochers, M. (1985). OPTIMAL ROUTING UNDER CAPACITY AND DISTANCE RESTRICTIONS. *Operations Research*. https://doi.org/10.1287/opre.33.5.1050

Laporte, G., Nobert, Y., & Taillefer, S. (1988). SOLVING A FAMILY OF MULTI-DEPOT VEHICLE ROUTING AND LOCATION-ROUTING PROBLEMS. *Transportation Science*. https://doi.org/10.1287/trsc.22.3.161

Meesuptaweekoon, K., & Chaovalitwongse, P. (2014). Dynamic Vehicle Routing Problem with Multiple Depots. *Engineering Journal*. https://doi.org/10.4186/ej.2014.18.4.135

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*. https://doi.org/10.1063/1.1699114

Mitrović-Minić, S., Krishnamurti, R., & Laporte, G. (2004). Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows. *Transportation Research Part B: Methodological*. https://doi.org/10.1016/j.trb.2003.09.001

Nagy, G., & Salhi, S. (2005). Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *European Journal of Operational Research*. https://doi.org/10.1016/j.ejor.2002.11.003

Okhrin, I., & Richter, K. (2008). The Real-Time Vehicle Routing Problem. In *Operations Research Proceedings 2007*. https://doi.org/10.1007/978-3-540-77903-2_22

Renaud, J., Boctor, F. F., & Laporte, G. (1996). A fast composite heuristic for the symmetric traveling salesman problem. *INFORMS Journal on Computing*. https://doi.org/10.1287/ijoc.8.2.134

Timmermann, T., & Schumanna, R. (2008). An approach to solve the multi depot vehicle routing problem with time windows (MDVRPTW) in static and dynamic scenarios. *KI 2008 - 31st*

Annual German Conference on Artificial Intelligence, Proceedings of the 22nd Workshop on Planen, Scheduling Und Konfigurieren, Entwerfen, PuK 2008.

Tjahjono, B., Esplugues, C., Ares, E., & Pelaez, G. (2017). What does Industry 4.0 mean to Supply Chain? *Procedia Manufacturing*. https://doi.org/10.1016/j.promfg.2017.09.191

Toth, P., & Vigo, D. (2002). Models, relaxations and exact approaches for the capacitated vehicle routing problem. *Discrete Applied Mathematics*. https://doi.org/10.1016/S0166-218X(01)00351-1

Voudouris, C. (1998). Guided Local Search - An illustrative example in function optimisation. *BT Technology Journal*. https://doi.org/10.1023/A:1009665513140

Wu, T. H., Low, C., & Bai, J. W. (2002). Heuristic solutions to multi-depot location-routing problems. *Computers and Operations Research*. https://doi.org/10.1016/S0305-0548(01)00038-7

Xu, H., Pu, P., Duan, F., & Hendy, A. S. (2018). A Hybrid Ant Colony Optimization for Dynamic Multidepot Vehicle Routing Problem. *Discrete Dynamics in Nature and Society*. https://doi.org/10.1155/2018/3624728

Yang, Z., van Osta, J. P., van Veen, B., van Krevelen, R., van Klaveren, R., Stam, A., … Emmerich, M. (2017). Dynamic vehicle routing with time windows in theory and practice. *Natural Computing*. https://doi.org/10.1007/s11047-016-9550-9