Doctoral Thesis

Heuristic search for the solution of environmental VRP incorporating vehicle load

Author:
Norfaieqah Ahmad

Supervisors:
Prof. Said Salhi
Dr. Adolf Acquaye

A thesis submitted in fulfilment of the requirements
for the degree of Doctor of Philosophy
in
Operational Research

Kent Business School

University of Kent

March 2018

Synopsis

Nowadays, there is an increasing concern with the environmental sustainability of human activities and current logistics practices. Various organisations and businesses are starting to pay more attention to their current logistics activities that may bring negative effects on the environment. Therefore, attempts are being made to develop route planning approaches that give the minimum impact to environmental damages. This research specifically seeks to design vehicle routes that serve a set of customers with the objective to minimise both the transportation cost and CO_2 emissions through the fuel consumption measure. The greenhouse gas emissions which cause global warming can be captured by the measurement of the amount of fuel consumed by a vehicle. This measure is influenced by numerous factors such as travelling distance, vehicle load, speed, acceleration, and road gradient.

This thesis presents an investigation into the impact of the load factor in the heuristic search for the case of environmental routing problem. In other words, this work examines the classical vehicle routing problem (VRP) namely the capacitated VRP (CVRP) with the incorporation of the load factor. We explore three types of heuristic searches which include the constructive heuristic, the local searches, and the integrated heuristics such as the composite and the multi-level metaheuristics. Newly developed formulations based on the load factor are introduced for the selected constructive method and the local searches. In the integrated heuristic, multiple local search operators are exercised all together within the search as it leads to a better exploration of the solution space. Furthermore, this research introduces a learning mechanism to guide the search more efficiently and in an adaptive way. Interesting results are obtained when using the 46 well-known VRP instances ranging in size from 21 to 1200 customers. It was found that in general, for small and large data sets, the learning search within the composite method yielded a better solution. However, for the medium data set, the learning search within the multi-level heuristic performed better.

The contributions of this study are fourfold: (i) the modification of the constructive heuristic namely the savings method by introducing the load factor, (ii) the incorporation of the load factor into several local searches, (iii) the exploration of the composite and multi-level metaheuristics, and (iv) the introduction of a learning mechanism into the overall search.

This thesis is arranged into seven chapters. A brief description of each chapter is given in the following. Chapter 1 provides an overview of the routing problems focusing on transportation planning. Issues associated with CO_2 emissions and its relationship with the load factor are also highlighted here. In addition, a common research methodology to solve the routing problems is discussed.

Chapter 2 investigates some of the literature related to this research. This includes an overview of VRP especially the capacitated one and its basic optimisation model. Some review of researches from the area of VRP focused on the one that associated with sustainable transportation activities are covered. Another class of VRP named the green VRP is explored which includes pollution routing problem and VRP with reverse logistic. We also studied some related literature to the estimation of vehicle fuel consumption and emission. This estimation method is essential in this research as we adopted it to approximate the total routing cost in the subsequent chapters.

Thereafter in Chapter 3, one of the most studied constructive heuristics namely, Clarke and Wright savings is analysed. We introduced two new savings formulations to incorporate the load factor known as the integrated and the combined savings. The emission model known as the Comprehensive Modal Emission Model (CMEM) is customised to be the cost function to calculate fuel consumption in evaluating the CO_2 emission. To put an emphasis on the load factor, we also conducted an assessment to use both directions of the established route. This assessment, which we refer to as the evaluation of the reverse route, remained to be of the essence throughout this research. Computational experiments performed using the 46 well-known VRP instances generated stimulating results.

Subsequently, Chapter 4 examines the first three basic local search operators which are the 2-opt, (1,0) relocate and (2,0) relocate. The computation of these operators is generally simple and relatively quick. The 2-opt is performed only for the intra-route case, but the other operators are used for both the intra-route and the inter-route cases. This chapter presents new mathematical formulations which cater for the load factor. An illustrative example is then provided followed by full

computational results obtained based on the same VRP instances. In addition, we also highlighted the computational comparison between the brute-force and the new formula-based searches. Its finding strongly supports the development of the new formulations.

In Chapter 5, three more intensive local search operators namely, the (1,1), (2,1) and (2,2) interchange are analysed. These operators are considered as advanced as they require relatively much more computing effort. Both the intra-route and the inter-route cases are implemented separately for a single operator. Similar to what was studied in Chapter 4, the load factor is adapted to obtain new mathematical formulations. An illustrative example is also given. This is followed by an extensive computational experiment which yields interesting outcomes.

Chapter 6 analyses the performance when all these local search operators are combined all together in the search. The method, which is referred to as the integrated search, comprised of two powerful mechanisms namely, the composite and the multi-level heuristics. These two metaheuristics refine the solution by using improvement schemes, also known as the local searches operators. These are arranged in a certain order which can be critical. The two methods are quite similar to each other, apart from the way it directed the search once a particular improvement scheme is implemented. In this study, we provided a brief description of the two methods as well as their algorithms which led to twelve variants. This is followed by the computational results conducted over all the variants. Two of these variants are found to be prominent and they are selected for the next stage where the learning mechanism is then introduced. We established a pseudo-random selection scheme based on a certain test probability to select the next local search. Three sets of test probability are chosen to be incorporated in the integrated search. Computational tests performed over five runs lead to interesting results.

Finally, Chapter 7 presents our conclusions and highlights some future research opportunities which we believe to be worthwhile.

Acknowledgements

In the Name of Allah – the Most Compassionate, Most Merciful.

The most significant thanks must go to my main supervisor, Professor Said Salhi. His direction and suggestions have been insightful, and I could not have completed this research and thesis without his continuous support, humour, patience and general encouragement.

My sincere thanks go to my co-supervisor, Dr Adolf Acquaye for his useful comments and suggestions.

I would like to formally thank the Ministry of Higher Education Malaysia and International Islamic University Malaysia for giving me the opportunity to undergo this journey and most importantly, offer me the sponsorship.

I am glad that I was surrounded by nice persons while working on the research and thesis at Kent Business School. Thank you to my fellow PhD researchers and also the administrative staffs. In addition, many thanks to my friend, Yvonne Sze Jeeu Fong who is very kind in giving me assistance and advice.

A big special appreciation to my beloved husband, Nik Aryadi for his daily encouragement and love. Not to forget my lovely daughter, Nik Arianna who is well-behaved at most times while I'm working on the thesis. I am also grateful to my parents and family for their support.

Lastly, I would like to extend my final credits to all who may have directly or indirectly contributed to the accomplishment of this work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Research Methodology for Environmental Vehicle Routing Problems

1.1 Introduction

This chapter provides an overview of routing problems focusing in the environment aspect by using the load factor. A brief of common research methodologies in solving the routing problems especially those techniques that will be used in this research is given. This is followed by the aim and the organisation of the thesis.

1.2 Environmental effect in transportation

Although transportation activities support increasing mobility demands for passenger and freight, it is also recognised as causing harm to our environment and our health. The hazardous impacts of transportation on the environment relate to climate change, air quality, noise, water quality, soil quality, biodiversity and land take (Rodrigue et al., 2017). Recently, nations worldwide increased their awareness that they need to address these hazardous impacts in order to achieve a sustainable transportation system.

At another level, transport activities also contribute to release greenhouse gases of which CO_2 is a potent element. The greenhouse gases consist of seven gases which are carbon dioxide, methane, nitrous oxide, hydrofluorocarbons, perfluorocarbons,

sulphur hexafluoride and nitrogen trifluoride. The last four gases are collectively referred as fluorinated gases or F-gases (DECC, 2015).

High concentration of CO_2 in the atmosphere poses a significant threat to the global environment. The emissions of CO_2 are directly proportional to the quantity of fuel consumed by vehicles. The vehicle fuel consumption largely relies on a variety of items including the type and size of vehicles, environment, and traffic-related parameters such as vehicle speed, load, acceleration and road gradient (Demir, Bektaş, and Laporte, 2011).

In this research, one of the main factors that affects fuel consumption, namely the vehicle load, is studied. In other words, the more load the vehicle carries, the more horsepower the engine demands. This leads to a higher fuel consumption. Vehicle load also influences the inertia force, rolling resistance and road slope force. Therefore, it is such a significant element of routing decisions that it cannot be ignored.

1.3 Routing problems in transportation planning

Routing problems can be identified as the development of a collection of tours on a network with the need to satisfy certain objectives without violating side constraints. These problems are performed in various kinds of networks including telephone networks, electronic data networks, and transportation networks. For transportation, it has become one of the most important means in everyday life that enables a particular entity to move from one point to another. The entity can be a person travelling or merchandise that is being distributed. The transportation sector is also regarded as the central infrastructure for economic and industrial growth and development for it mainly focuses on the maximum profit at hands and on the reduction of their operational costs.

Relating to that, the majority of road freight transportation is operated by trucks, which mostly run on diesel engines and which emit a large portion of pollutants. Diesel engines have been known as the main sources of nitrogen oxides (N_2O), particulate matter (PM) and carbon dioxides (CO_2) emissions which are associated to a series of health problems when continuously exposed to them (Demir, Bektaş, and Laporte, 2014a).

The vehicle routing problem (VRP) is fundamental in road transportation planning and often aims at minimizing the cost of the routes. VRP can be defined as the design of least cost routes for a fleet of vehicles in satisfying the demand of a set of customers under certain constraints. The vehicle begins and ends the journey at its own depot and each customer must be visited once only. This problem could be enriched by the incorporation of terms related to fuel consumption to meet the environmental concerns. Therefore, consideration of the load factor within the formulation in developing optimal or good routes is the purpose of our study.

Toth and Vigo (2002b) examined VRP and some of its main variants along with exact and heuristics methods developed for solving it as well as its application in real-world instances. These two types of optimisation methodologies will be briefly reviewed in Subsection 1.4. Figure 1.1 shows the basic version of VRP and their interconnections with the addition of green VRP variant in the shaded box as most studies under this variant adapted the existing VRP model to integrate environmental concerns in addition to economic interests. This VRP variant will be discussed further in Chapter 2.

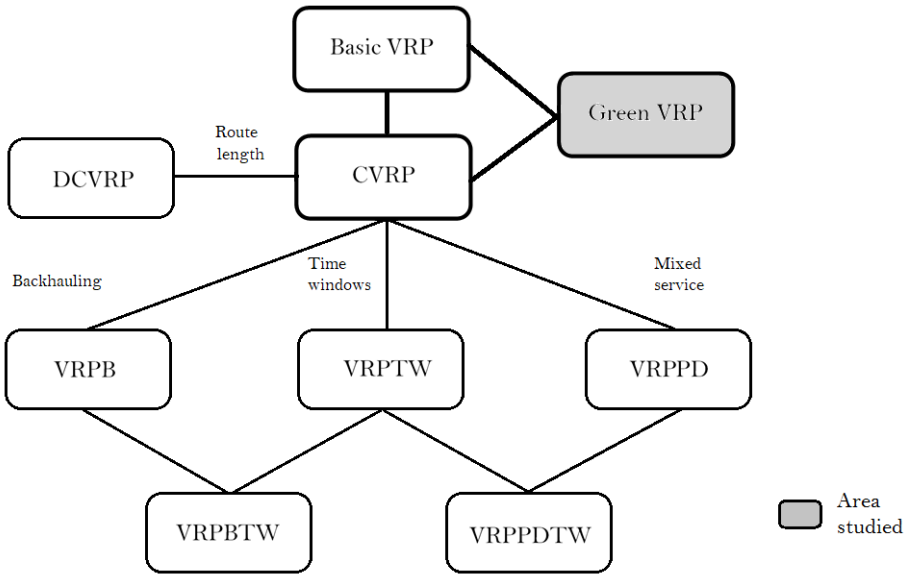

FIGURE 1.1: Basic problem of VRP class and their interconnections (Toth and Vigo, 2002b)

1.4 Research Methodology

Generally, methods to solve optimisation problems are categorised in two groups which are the exact and the approximate/heuristic methods. The exact methods are classical approaches to solve combinatorial and global optimisation problems optimally. They usually include, for example, branch and bound, cutting plane, and dynamic programming. The idea is to recursively split up a problem into basic sub-problems so that they can be solved at once. Then, its solutions are combined to form a solution to the original problem. Though the exact methods are used to produce an optimal solution, they can be extremely time-consuming when solving large real-world combinatorial or global optimisation problems. From the literature, it is found that the exact methods can efficiently provide a solution for a VRP problem with up to about 100 customers. This leads to the vast research in heuristic design especially for the one with a large-scale and complex practical problem that are inappropriate for exact methods. Some of the relevant heuristics will be briefly discussed here but for more information see the recent book by Salhi (2017).

There is no general basis for devising a good heuristic method and it is not certain for any problem to assuredly obtain a good quality solution. The success of the heuristic method depends upon its aptitude to adjust to a particular realization, to avoid entrapment at local optima, and to manipulate the basic structure of the problem. Based on these notions, numerous heuristic search methods have been fostered to tackle difficult combinatorial and global optimisation problems. The search seeks to approximately produce a good-quality solution within reasonable computing times and efforts that can be considered to be acceptable for practical purposes.

Heuristic search can be classified in numerous ways. One categorization, consisting of four groups, is provided by Salhi (2017) as follows:

(i) improving solutions only

(ii) not necessarily improving solutions

(iii) population-based

(iv) hybridisation

In this research, we mainly focused on the heuristics in the first category (i). For our purpose, we further divided it in three types including the constructive heuristics, the local search, and metaheuristics. This section is followed by a short description of multi-objective optimisation method that will also be reviewed in the next chapter. In the following subsections, we briefly describe the methods that we use in this study though others that belong to the same category are also briefly mentioned here.

1.4.1 Constructive heuristics

This approach is often employed to generate an initial solution. It is performed by constructing the solution starting from scratch. Certain selection strategies are applied during the build-up of the solution. This will then contribute to the final solution. We give a brief description of four VRP-based constructive heuristics with the first two being explored in this thesis, either to construct the solution or to refine it.

Techniques used

(a) Savings method

One of the earliest method and frequently use in practice to solve the VRP is the Clarke and Wright (1964) savings. It was introduced with a simple but yet an effective procedure to produce a near-optimal solution. In this study, we implement this method to generate our initial solution and incorporate the load factor in its algorithm. We shall discuss this method and its algorithm in more details in Chapter 3.

(b) Insertion-based method

Another intuitive approach to the VRP is to start with a sub-route and extend it by adding the unserved customers one by one until all customers belong to a route. The rule of inserting a new customer is to choose the point in which the insertion would increase the cost the least in the route cost. Mathematically, the insertion cost of unrouted customer k between customers i and j can be calculated as

$$I(k, ij) = d_{ik} + d_{kj} - d_{ij}$$

This approach is flexible as it can also be applied to improve the initial solutions. It can be used as we will show in subsequent chapters. The improvement can be used within the route (intra-route) or among the routes (inter-route). In this study, we refer to this as μ-relocate which will be explained next in the local search, Subsection 1.4.2.

Other techniques

(a) Nearest neighbour search

This method is also one of the earliest and simple to conduct. It is used to solve the travelling salesman problem (TSP) where the salesman makes the first visit to one city, either randomly or chosen using a certain criterion, and goes on to the next nearest unvisited city. This process is repeated until all cities are visited. Similarly, this is applied for the VRP in which the tour will start from the depot and the nearest customer to it. This customer then becomes a reference point for the next customer, who is unvisited and the nearest to the reference point, to be visited. This process continues as long as the vehicle capacity or the distance constraints are not violated. Otherwise, a new tour is created and the process recurs until all customers are assigned.

(b) Sweep method

This approach is proposed by Gillett and Miller (1974) where an iterative procedure which is similar concept to the nearest neighbour method is used to construct the routes. The selection of the first customer k in the route can be crucial. For example, a customer with the largest demand can be visited first. Afterwards, instead of selecting the nearest customer, an unrouted customer with the smallest angle from the depot to k is inserted in the route. The search progressively adding customers into the route until a route is full. This means that inserting new customer in the route will violate the constraints. Consequently, the resulted routes are non-intersecting, although more routes than necessary may be developed.

1.4.2 Local search methods

Local search methods, also known as improvement heuristics, are implemented on an established solution to improve the solution quality. We provide some

definitions and description of the notions related to these methods, as given in Salhi (2017).

Local search definition The mechanism (i.e. operator) to generate a solution X' from a solution X in a given neighbourhood $N(X)$ where $X' = Argmin\{F(X); X \in N(X)\}$.

Neighbourhood definition A solution, say $X \in S$ in which S is the set of feasible solutions, has an accompanying set of neighbours, say $N(X) \subset S$. Therefore, $N(X)$ is the neighbourhood of X.

Selection strategies The following strategies are often used in selecting a solution X' from a solution X in a given neighbourhood $N(X)$.

(i) Best improvement strategy, which analyses all solutions $X' \in N(X)$ in the neighbourhood and accepts the attribute that returns the best outcome based on a particular acceptance condition (i.e. the least cost in a minimisation problem).

(ii) First improvement strategy, which selects the first solution $X' \in N(X)$ once the acceptance condition is satisfied.

Local search operators In this study, three types of local search operators are implemented to improve the initial solution. We investigate the effect of the load factor by modifying the formulation of these operators.

(a) The λ-*opt* method is originally proposed by Lin (1965) where he introduced the λ-optimality concept. Basically, the method is implemented by taking out λ arcs from the route and then reconnecting the λ remaining segments in a feasible way. The procedure halts at a local minimum when further improvements cannot be obtained. In this research, we consider $\lambda = 2$ and present our investigation in Chapter 4. Other values of λ could also be attempted.

(b) The μ-*relocate* shifts a number of customers μ from a route to a different position in the same route (i.e. intra-route) or to another different route

(i.e. inter-route). The most common operator of this type is 1-relocate or sometimes called the (1,0) operator, where one customer is removed from a route and inserted in another location. This type of operator is widely used in the research due to its simplicity and time-saving computation. This operator could be easily extended to a more complex computation depending on the value of μ. For instance, if $\mu = 2$, the two customers could shift either separately or jointly to another location. In this study, we consider $\mu = 1, 2$ and the customers are being shifted consecutively when $\mu = 2$. Further exploration is given in Chapter 4.

(c) The μ-ψ *interchange* or also known as (μ,ψ) operator exchanges μ customer from one route with ψ customer from another route. Here, the exchange position is not necessarily the same as before. In other words, once μ customers are removed from route, say i, ψ customers from route j can be inserted anywhere in route i, and vice versa. This version is referred to as the best exchange which will be discussed further in Chapter 5. If ψ customer is inserted in the previous position of μ customers, the procedure reduces to a simpler version which we refer to as the direct exchange or the swap move. This operator could also be implemented for both the intra- and the inter-route cases. We apply the direct exchange or swap move and the best exchange for the intra- and the inter-route cases, respectively. A similar implementation could also be applied for other values of μ and ψ. The illustration of how this mechanism can be used is also shown in Osman (1993).

1.4.3 Metaheuristics

Metaheuristics is another class of heuristics which exploits classical heuristics with refined strategies to explore further into the solution space in identifying a good solution. Osman and Laporte (1996) in their bibliography provide a formal definition of metaheuristic as given below:

> *An iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space, learning strategies are used to structure information in order to find efficiently near-optimal solutions.*

In this study, we concentrate on two powerful mechanisms which are the composite and multi-level heuristics. A general idea of these two methods are presented briefly below. Other metaheuristics are also briefly mentioned, though they are not investigated in this work, which could be worth to explore if necessary.

Simple composite heuristic

This method uses an improvement procedure or local search one after another to generate better quality solutions. One possible skeleton of this approach is given below in Salhi (2017). We shall examine this method further in Chapter 6.

Step 1: Initialise k possible refinements (usually $k = 2$ or 3) and generate an initial solution.

Step 2: Apply the k refinements in sequence on the incumbent solution.

Step 3: If there is improvement, repeat Step 2. Otherwise, record the last solution and stop.

Multi-level heuristic

This mechanism can be considered as an extension to the composite method. This is because, in each level, a single local search or even a composite heuristic itself could be operated. The multi-level mechanism is designed to allow for as much distortion as possible to the solution to provide diversity on condition that the computing effort remains acceptable. A general algorithm of this method, which is originally proposed by Salhi and Sari (1997) with p levels is presented below. For clarity, a Type p composite heuristic is considered less complex than Type $p + 1$ ($p > 1$). Further investigation of this method is given in Chapter 6.

Level 1: Generate an initial solution.

Level 2: Apply a composite heuristic of Type I.

Level 3: Apply a different composite heuristic of Type II. If there is improvement, go to Level 2.

.

.

.

Level p: Apply a composite heuristic of Type $p-1$. If there is improvement, go to Level 2.

Level $p+1$: If none of the stopping criteria is met, perform a diversification mechanism and go back to Level 2. Else, record the final solution and stop.

A learning strategy is also introduced and implemented with the conjunction of the above two methods. This is done to enhance and intelligently guide the search to be more adaptive and practically more effective. This will also be presented in Chapter 6.

Other metaheuristics

There are many methods that fall into this category. Among others, a variable neighbourhood search (VNS), which is firstly proposed by Mladenović and Hansen (1997), seeks to systematically change the neighbourhoods followed by a local search, though not always. Simulated annealing (SA) originally developed by Metropolis et al. (1953) comes from an analogy of the cooling process of a material. The algorithm follows the procedure to gradually reduce the system temperature until it turns into a steady, frozen state. Like SA, tabu search (TS) method accepts non-improving moves but in a deterministic way as it explores the entire spaces to escape from local optimality. A 'tabu' or prohibition is therefore introduced to prevent cycling (i.e. the search from coming back to the attributes that recently involved in the procedure). This is developed by Glover (1986). Genetic algorithm (GA) is a population-based heuristic put forward by Holland (1992). The new population is constructed from the former 'parent' by combining its features through crossover and mutation operators. For more information on these methods and others, see Salhi (2017).

Multi-objective optimisation

Multi-objective optimisation simultaneously solves two or more conflicting objectives in order to obtain non-dominated solutions. In this study, we adapted this method to formulate the objectives to minimise the total distance and vehicle fuel consumption. A weighting method, which assigns a non-negative weighting coefficient to each objective function, aims to minimise the weighted sum of the

objectives. More details on this area, which will be used in this research will also be reviewed in Section 2.4.

1.5 Aim and organisation of the thesis

The aim of this research is to investigate the effect of vehicle load in evaluating the amount of fuel consumption in a comprehensive way. This led to new mathematical formulations for constructive heuristic and local searches to cater for the load factor in the area of routing. To our knowledge, these are new formulations produced for the first time. This contribution adds a useful tool to deal with environmental effect which is crucial in nowadays.

This thesis is presented in seven chapters. This chapter has introduced a related subject to transportation routing and the environmental concern of this activity. In addition, a common research methodology is also discussed. In the next chapter, some of the literature related to this research are examined, including a review of research from the area of vehicle routing problems associated with sustainable transportation activities and the estimation of vehicle fuel consumption and emission.

A constructive heuristic for generating an initial solution of the problems is discussed in Chapter 3. This chapter also covers the evaluation procedures in adapting the load factor in the routing problem. Chapters 4 and 5 describe the modification of the local search operators based on the load factor. There are six operators in consideration for this matter. An illustrative example along with the computational experiments' result for each operator are also discussed. Chapter 6 analyses two metaheuristic searches, namely the composite and multi-level heuristics to observe the performance of using multiple operators within the search. In addition, a learning mechanism is also introduced. Conclusions are drawn and suggestions for future research are given in the final chapter.

Chapter 2

Review of Green VRP and Related Literature

2.1 Introduction

This literature review examines previous studies that have been undertaken in the area of vehicle routing problem (VRP). Several VRP variants especially the capacitated VRP and its basic optimisation model are briefly described. We discuss the class of VRP named the green VRP which relates to sustainable transportation practices from operations research perspectives. As this research combines both cost and CO_2 emissions, some research works concerning the multi-objective optimisation problem are also covered. Models for assessing fuel consumption and vehicle emission are also considered.

2.2 Overview of vehicle routing problem

Vehicle routing problem (VRP) has arisen more than fifty years ago when Dantzig and Ramser (1959) introduced the problem as the truck dispatching problem where it is concerned with routing a fleet of gasoline delivery truck between the terminal and a large set of service stations provided by the terminal. The aim is to find a way so that the truck is assigned to stations and thus satisfying its demand while keeping the total mileage of the fleet at the minimum.

In this study, we focus on the problems pertaining to the distribution of goods between depots and final user (customers). These problems are mostly called as

Vehicle Routing Problems (VRPs) or Vehicle Scheduling Problems. The distribution of goods involves the service under a given time interval, for a set of customers by a set of vehicles that are situated in one or more depots. The vehicles are operated by a set of staffs (drivers) that travel through an appropriate road network. Generally, VRP requires the solution by constructing a set of routes, each route operated by a vehicle that begins and ends its tour at its own depot. Apart from that, the solution must ensure that all the customers' requirements are met, all the operational constraints are accomplished, and the overall transportation cost is minimised. More details of the VRP terminology can be found in Toth and Vigo (2002a).

2.2.1 The Capacitated VRP

Researchers have come out with many other variants of VRP and proposed numerous solution methods. The simplest version of VRP which is mostly studied in the literature is when the amount of transported goods is restricted to the vehicle capacity. This problem is called the Capacitated VRP (CVRP). In the CVRP, all the customers correspond to delivery services along with the predetermined and indivisible demands (the number of goods to be delivered to them). The vehicles used are homogenous and located at one main depot. Only the capacity limitations are imposed on the vehicles. The objective is to serve all the customers with the minimisation of the total cost of the routes.

The CVRP may be described through the graph-theoretic problem where on an undirected graph $G = (V, A)$ there exists a set $V = \{0, 1, \ldots, n\}$ of vertices together with a set $A = \{(i, j) : i, j \in V, i \neq j\}$ of arcs. The graph represents the road network used for the transportation of goods with arcs represent the road sections and with vertices stand for the depot and customers' location. Vertex 0 denotes a depot and sometimes it is expressed as vertex $n + 1$.

A set of K identical vehicles with an individual capacity Q is accessible at the depot. Each customer $i \in V \setminus \{0\}$ is associated with a non-negative demand $q_i \leq Q$ that should not exceed vehicle capacity to ensure feasibility. However, in case it does, a full vehicle load will be used.

A non-negative cost, c_{ij} is defined on the arc set A and represents the cost incurred when travelling from vertex i to vertex j. The cost is associated with the use of the vehicle which can be found in the unit of distance, travel time, etc. If G is

a directed graph, the cost matrix c is asymmetric, then this problem is known as asymmetric CVRP. Else, $c_{ij} = c_{ji}$ is existing for all $(i, j) \in A$ and the problem is known as symmetric CVRP.

In this study, the solution method is tested for the symmetric VRP as the cost c_{ij} is defined as the Euclidean distance between the two vertices i and j. Later, it is compared to the one with the consideration of varying load in c_{ij} function which led to the problem becomes asymmetric. Further details can be found in Chapter 3. Additionally, the summary of the most important concepts, algorithms and results for the classical VRP in which only vehicle capacity constraints are present can be referred in Laporte (2007).

Among the previous works that successfully solved the CVRP using the VRP benchmark instances with high quality outcomes include Xu and Kelly (1996), Li, Golden, and Wasil (2005), Tarantilis (2005), Pisinger and Ropke (2007), Chen, Huang, and Dong (2010), Zachariadis and Kiranoudis (2010), and Jin, Crainic, and Løkketangen (2012).

Basic optimisation model of CVRP

There are three main mathematical programming formulations that can be used to model the VRP. The first modelling approach known as vehicle flow formulations make use of integer variables that associated with each arc or edge of the graph. The variables add up the number of times the arc or edge is traversed by a vehicle. This model is often used for a basic version of VRP. It is suitable for cases in which the solution cost can be expressed as the sum of any costs associated with the arcs. The formulations are presented as follows.

The second approach is based on the commodity flow formulations where additional integer variables are associated with the arcs or edges. The variables correspond to the flow of the commodities through the vehicle routes. This model has been known as a basis for the exact solution of CVRP.

The third approach is called as a set partitioning problem which uses an exponential number of binary variables associated with a different feasible circuit. The VRP is then expressed as a set partitioning problem with the objective to serve all the customers with the minimisation of the collection of circuits and satisfy additional side constraints. An example of the use of this approach can be found in Salhi, Wassan, and Hajarat (2013).

Vehicle flow formulation For the use of this study, here we describe an integer linear programming formulation for basic CVRP. The model is presented in Toth and Vigo (2002a) known as two-index vehicle flow formulation. In this model, the variable x is used to show if a vehicle traverses an arc in the optimal solution. That is if arc $(i, j) \in A$ belongs to the optimal solution then the variable x_{ij} takes value 1 and 0 otherwise.

$$\min \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij}$$

subject to

$$\sum_{i \in V} x_{ij} = 1 \qquad \forall j \in V \setminus \{0\}, \tag{2.1}$$

$$\sum_{j \in V} x_{ij} = 1 \qquad \forall i \in V \setminus \{0\}, \tag{2.2}$$

$$\sum_{i \in V} x_{i0} = K, \tag{2.3}$$

$$\sum_{j \in V} x_{0j} = K, \tag{2.4}$$

$$\sum_{i \notin S} \sum_{j \in S} x_{ij} \geq r(S) \qquad \forall S \subseteq V \setminus \{0\}, S \neq \emptyset, \tag{2.5}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i, j \in V. \tag{2.6}$$

The indegree and outdegree constraints (2.1) and (2.2) require that only one arc arrives and leaves each vertex associated with a customer, respectively. Constraints (2.3) and (2.4) impose the degree requirements for the depot vertex. The capacity-cut constraints of (2.5) impose both the connectivity of the solution and the vehicle capacity requirements. In fact, they specify that each cut $(V \setminus S, S)$ defined by a customer set S is crossed by a number of arcs not smaller than $r(S)$ (minimum number of vehicles needed to serve set S).

2.2.2 Other VRP variants

The CVRP has been extended to a wide range of VRP versions. The first extended version is when the capacity constraint for each route is replaced by a route length restriction. This is called Distance-Constrained VRP (DCVRP) In this case, a non-negative length, t_{ij} of each arc $(i, j) \in A$ is accumulated such that for each

route, the total length cannot exceed the maximum route length, T. Furthermore, if arc lengths represent travel times, a service time, s_i is associated with each customer i indicating the time duration of servicing the customer at its location.

Another common version is the VRP with Time Windows (VRPTW) where under the vehicle capacity constraints, each customer i must be visited within a predefined time interval $[a_i, b_i]$, named a time window. The time in which the vehicles travelling from depot to customer location, travel time t_{ij} for each arc $(i, j) \in A$ and an additional service time s_i for each customer i are also provided. The vehicle must begin to serve the customer within the associated time window and stop at their location for s_i duration. The importance of applications of VRPTW in real cases has made researchers do efforts to find appropriate algorithms and propose different methods to solve them. For example, Figliozzi (2012) presents an algorithm that can tackle time dependent vehicle routing problems with hard or soft time windows without any alteration in its structure and introduced readily replicable test problems that capture the varying speed of congested urban settings.

VRP with Backhaul (VRPB) is an extension of CVRP where it involves two subsets generated from the customers' set $V \setminus \{0\}$. The first subset, L includes linehaul customers, each calls for a given amount of goods to be delivered to them. The second subset, B comprises of backhaul customers each requiring a given amount of goods to be picked up. The solution for VRPB must observe that a set of routes is constructed with the minimum cost wherein each route, all the deliveries must be made before any pickups and the vehicle capacity constraints is not violated by either the linehaul or backhaul customers assigned to the route.

VRP with Pickup and Delivery (VRPPD) is a VRP variant which requires a known amount of goods to be delivered and at the same time, may be picked up at customer i. A basic version of this problem assumed that at each customer, the delivery is performed first. Thus, the current vehicle load must be updated at each point and this must be non-negative and cannot exceed the vehicle capacity Q. This problem is beneficial to put the vehicle at full use when it returns to the depot after all deliveries have been carried out. For more information on this topic, see Wassan and Nagy (2014).

Both VRPB and VRPPD are in the field of delivery and pickup problems. The first problem, VRPB usually deals with the transportation of goods between customers and the depot. The second problem, VRPPD mostly refers to the situations where

goods are transported between the pickup and delivery locations (customers). A two-part survey paper on pickup and delivery problems by Parragh, Doerner, and Hartl (2008a) and Parragh, Doerner, and Hartl (2008b) provide a clear classification between VRPB and VRPPD and their variants.

Vehicle routing problem has also been extended and modified to cater for real-world problems. These include the addition of a variety of constraints to be imposed in terms of vehicle, driver, customer, type of goods, and the environment.

2.3 Green vehicle routing problem

Government and business organisation have recently paid a close attention to take the environmental, ecological and social effects into consideration when constructing logistics policies, in addition to the economic benefit. Green logistics is introduced knowing that our production and distribution logistics policies in the present are not lasting for future generations. Activities promoting green logistics include measuring the environmental impact of various distribution strategies, reducing the energy usage in logistics activities as well as reducing waste and managing its treatment (Sbihi and Eglese, 2009). A review by Dekker, Bloemhof, and Mallidis (2012) highlighted the contribution of operations research in integrating the environmental aspects in logistics.

In order to achieve the environmental-friendly logistics policy, transportation scheme also needs a change toward sustainable distribution network with the least damageable to the environment as transportation features in major parts of logistics. Björklund (2011) has provided an alternative definition of green transportation services as a service that has a lesser or reduced negative impact on human health and the natural environment when compared with competing transportation services that serve the same purpose.

The issues concerning green transportation are wide and involve participation of whole division from local passenger to the high authority policy-maker. For example, the promotion of alternative fuels utilisation, modern technology electronic vehicle, green intelligent transportation systems, and other environmentally friendly infrastructures. The incorporation of the environmental themes in routing problems can be considered as green transportation.

All this while, a number of VRP studies has been carried out with the aims at minimising the total distance travelled and total assigned vehicles, it has already satisfied the effort of sustainable transportation through reducing consumption level and subsequently reducing the CO_2 emissions from road transportation (Salimifard, Shahbandarzadeh, and Raeesi, 2012). Nevertheless, researchers have often been unconscious of their indirect contributions to the environment. The need to properly address the environmental issues specifically in routing problems, such as minimizing the CO_2 emissions while travelling has just arisen to become one of the hot topics in the literature in recent years.

Studies of VRP that is in relation to sustainable transportation issues from the operational research perspectives are denoted as green vehicle routing problems. Green VRP is concerned with the optimisation of energy usage in transportation routing. High volumes of energy usage and air pollution pose a threat on our ecological environment. The greater concern to transportation companies in pursuing fewer fuel costs is the amount of fuel consumed itself (Xiao et al., 2012). Fuel cost attributes a significant part of the total operational transportation cost. Thus, reducing fuel consumption and shifting it for efficient transportation strategies would lead to a beneficial course of action not only for economic gain but also for environmental savings as the emission of CO_2 can be correspondingly lessen.

To conclude the amount of fuel consumed in the routing model, there are many factors that influenced the vehicle traveling cost along a route which are essential in quantifying the fuel consumption. These factors are categorised in two parts according to their relationship with travelling schedule (Xiao et al., 2012). The first one has a direct relationship with travelling schedule, such as distance, vehicle load, speed, road conditions, fuel consumption rate (fuel consumption per distance unit), and etc. While parameters such as depreciation of the tyres and vehicle, maintenance, driver wages, and government tax are grouped in the second set which has no direct relationship with travelling schedule. By laying down all the factors as mentioned above, the amount of fuel consumption can be computed by controlling the factors in the first set.

In a recent study by Demir, Bektaş, and Laporte (2014a), factors affecting fuel consumption are summarised and divided into five categories; vehicle, driver, environment, traffic, and operations. According to the authors, most fuel consumption formulation has focused on vehicle, environment, and traffic influences but do not include the driver related concerns which are rather difficult to measure. Apart

from that, operations related factors are regularly considered as externalities that influence the amount of fuel consumed. A related study by Kontovas (2014) explained the relationship of fuel consumption and emission, and how to estimate them in the area of ship routing and scheduling.

There is a shortage of published work of VRP in determining the amount of fuel at minimum use. The so-called Energy Minimising Vehicle Routing Problem by Kara, Kara, and Yetis (2007) considered vehicle load and distance travelled in their new cost function that mainly focuses on the total energy consumption of the vehicle. By reducing the total work done by the vehicle, their aim is to consequently minimise the total energy used in terms of fuel consumption. The CVRP problem is formulated using integer programming for collection and delivery cases. Their comparative analysis between distance-minimising and energy-minimising solutions shows that there is a trade-off such that the energy usage increases as the total distance decreases. A related study by Pradenas, Oportus, and Parada (2013) defined a new objective function that considers minimising the required energy instead of the distance travelled in the VRPBTW.

Xiao et al. (2012) improve the existing current studies on the CVRP through a proper formulation of the work characteristics by taking fuel consumption rate as a load dependent function. A string-based simulated annealing algorithm is developed to solve the proposed model. The authors provided insight such that the pursuing of the shortest distance that leads to lower cost may not be efficient in reducing fuel consumption. This is because the vehicle load is also significant besides the distance in contributing to the overall fuel consumption along the routes. Gaur, Mudgal, and Singh (2013) consider both load and distance travelled to measure fuel consumption in their cumulative VRP model.

According to Kuo (2010), fuel consumption is regarded as an important index in the VRP as the need to address the environmental issues become significant particularly in reducing carbon emission. The author develope a model to calculate the total fuel consumption for time-dependent VRP (TDVRP) in which he assumed travel speed and travel times are depended on the departure time in routing problems. Transportation distance, speed and loading weight are also considered in calculating the model. A simulated annealing algorithm is proposed to optimise the fuel consumption of the routes. Based on the comparative analysis between fuel consumption minimisation and both distance and travel time minimisation, the result indicate that more than 20% improvement in fuel consumption can be

achieved. Later, Kuo and Wang (2011) publish a similar work where they propose a simple tabu search algorithm to solve the model.

Studies of Green VRP also deal with the use of modern vehicles and greener fuel sources other than petroleum-based fuels in routing problems. Business organisations are transforming their vehicle fleet to the use of alternative fuel vehicles (AFV) in order to lower the negative impact of the environment. Among the earliest work done to address this is Erdoğan and Miller-Hooks (2012) in which their objective is to find the shortest distance in distribution system using the AFVs. They develop formulation and solution technique to overcome the difficulties of limited refuelling infrastructure using a mixed integer linear programming and constructive heuristics with improvement technique. Later, Felipe et al. (2014) extend the study by introducing Green Vehicle Routing Problem with Multiple Technologies and Partial Recharges (GVRP-MTPR) using electric vehicles.

Another relevant study is carried out by a group of researchers in Serbia who developed a model in routing environmentally friendly vehicles (EFV) in urban areas (Ćirović, Pamučar, and Božanić, 2014). The EFV is assumed as 'clean' motor vehicle where it does not release harmful substances in its exhaust fumes and make less noise without triggering other environmental damages. The authors propose an adaptive neural network trained with a simulated annealing algorithm for a neuro-fuzzy model of vehicle routing to minimise air pollution, noise level and logistics operating costs. A similar model has been used by Jovanović, Pamučar, and Pejčić-Tarle (2014) in public transportation scenario.

In the next subsections, we first give some reviews about the new variant of routing problem under this topic which is the pollution-routing problem. Next, we also describe briefly about VRP in reverse logistic in literature which can be categorised in four areas; (a) selective pickups with pricing, (b) waste collection, (c) end-of-life goods collection, and (d) simultaneous distribution and collection.

2.3.1 Pollution-Routing Problem (PRP)

The Pollution-Routing Problem (PRP) aims at choosing a vehicle dispatching scheme with less pollution, particularly in minimising carbon emissions. The PRP model approximates the total used energy on a given road segment, which directly converts into fuel consumption and further into greenhouse gas emissions.

A number of studies aims at a reduction CO_2 emissions including for instance, the work done by Tiwari and Chang (2015) who consider a distance based approach to compute the CO_2 emission while adding the truck load as a factor for CO_2 emissions. A block recombination approach is applied to generate different clusters (block) for each city visited by a different truck. Then, the cluster with a higher fitness value is selected and combined it according to their rank.

Ubeda, Arcelus, and Faulin (2011) present the green logistics which concentrated at the operational level of transportation with the target to reduce the CO_2 emissions in the real case study of Eroski (food distribution company in Spain). Several adjustments to current logistics management practices are proposed. These aim to improve delivery routes scheduling, reduce the number of trips by adding backhaul movements, and develop an environmental friendly method to solve routing problems. The authors applied a distance-based method for calculating CO_2 emissions, which is based on distance travelled and distance-based emission factors.

On the one hand, a PhD thesis by Palmer (2007) investigated the efficient method in freight vehicle routes that minimise CO_2 emissions to the environment. The author develop a model that fits novel techniques in measuring the CO_2 emitted by a vehicle which has traffic using software tool that involved the use of digitized road network. Attributes such as traffic volume, road categories, speed, acceleration, and driving cycle are considered in the study which influenced vehicle fuel consumption and subsequently converted into CO_2 emissions.

There are many other works that consider CO_2 emission in determining vehicle routes to satisfy customer demand and vehicle capacity. However, not many emphasised on the vehicle load factors through the fuel savings methods. For instance, Kwon, Choi, and Lee (2013) studied the use of heterogeneous fixed fleet VRP with the highlight of carbon emission in optimising vehicle routes. Bauer, Bektaş, and Crainic (2009) emphasised minimising greenhouse gases emissions in the context of intermodal freight transportation. Soysal, Bloemhof-Ruwaard, and Bektaş (2015) modelled the time-dependent two-echelon capacitated vehicle routing problem (2E-CVVRP) with a comprehensive mixed integer linear programming formulation that comprises vehicle type, travelled distance, vehicle speed, load, multiple time zones and emissions.

Oberscheider et al. (2013) aimed to minimise driving times and greenhouse gas emissions in timber transport. They modelled the problem as a multi-depot vehicle routing problem with pickup and delivery and time windows (MDVRPPDTW).

Suzuki (2011) presented the truck routing problem with the objective to minimise fuel consumption and pollutant emissions using the traveling salesman problem with time windows (TSPTW) model. Yang et al. (2013) established a 0-1 mixed integer programming model to minimise carbon emissions for the airport shuttle service. Fagerholt, Laporte, and Norstad (2009) introduced speed optimisation procedure to reduce fuel emission in shipping scenarios which was later adapted in the majority of PRP studies.

Bektaş and Laporte (2011) extended the classical VRPTW by introducing the term pollution routing problem (PRP) which comprehends the travel distance as well as the amount of greenhouse emissions, fuel, travel times along with their costs in the objective function. The authors presented mathematical models for the PRP with and without time windows and performed computational experiments on realistic instances. Through the incorporation of fuel consumption and CO_2 emissions into existing planning methods for vehicle routing, this study has potential in yielding savings of the total cost.

Following on that, Demir, Bektaş, and Laporte (2012) proposed a new solution method to solve PRP using an adaptive large neighbourhood search heuristic algorithm while creating several new operators to improve their current solutions, and employed speed optimisation procedure. The authors then presented a multi-objective model where they adapted the previous solution approach with two objective functions concerning fuel consumption and driving times minimisation (Demir, Bektaş, and Laporte, 2014b).

Franceschetti et al. (2013) introduced time-dependent PRP which stretched the PRP by taking into account traffic congestion. The paper described a procedure to optimise departure times and speeds on a fixed route, also building on the analytical results proven for the single-arc version of the problem. Knowing the fact that vehicles with a higher power engine and larger loading capacity do not necessarily results in fuel savings, Koç et al. (2014) introduced the fleet size and mixed PRP wherein the use of heterogeneous vehicle fleet is considered. The aim is to keep the total of vehicle fixed cost and routing cost at a minimum where the routing cost comprising the fuel and CO_2 emissions cost and driver cost. The paper applied a hybrid evolutionary meta-heuristic to solve the problem and then demonstrated the significant differences of using a heterogeneous fleet over a homogeneous one.

A time window with pickup and delivery pollution routing problem (TWPDPRP)

is introduced by Tajik et al. (2014) to deal with uncertain information of transportation environment including fuel consumptions, CO_2 emissions cost and vehicles velocity. The paper included many factors that impacted fuel consumption and CO_2 emissions in their objective function of cost formulation. These include the physical condition of the chosen roads, the weight of the vehicles, and the loads they carry, frontal surface of the vehicles, friction rate of air, acceleration, and velocity of the vehicles. Furthermore, a robust concept of a mixed integer linear programming is introduced by applying the recent robust optimisation theory. Another recent study by Kramer et al. (2015) extends the PRP by applying a new matheuristic approach that is a hybrid iterated local search leading to the latest state-of-the-art outcomes.

2.3.2 VRP in reverse logistics (VRPRL)

Reverse logistics can be defined in many ways and has been widely studied across numerous logistics activities including transportation. Govindan, Soleimani, and Kannan (2014) described it as a process which starts from end users where used products are collected from customers and then attempts to manage end-of-life products through different decisions including recycling, remanufacturing, repairing, and finally, disposing of some used parts. VRP in reverse logistics is concerned with the management of the way goods have been transported with their recovered value after use. Huge amount of research on reverse logistics are found but from the VRP views, there are not many studies. Unlike the Lin et al. (2014b) survey, this review focuses on the works of reverse logistics in vehicle routing problem with clear environmental considerations. Based on the paper, the VRPRL is divided into four types, which are selective pickups with pricing, waste collection, end-of-life goods collection, and simultaneous distribution and collection. The following reviews employ their category in order to organise the existing research.

(a) Selective pickups with pricing Aras, Aksen, and Tuğrul Tekin (2011) optimised the routes of a homogeneous fleet of capacitated vehicles each of which will depart from a collection centre, visit a number of dealers to pick up cores, and return back to the same centre. Privé et al. (2005) can be modelled as a variant of the vehicle routing problem with a heterogeneous vehicle fleet, time windows, capacity and volume constraints. The objective function here is to combine routing costs and the revenue resulting from the sale of recyclable material.

(b) Waste collection Apaydin and Gonullu (2008) and Zsigraiova, Semiao, and Beijoco (2013) benefit the use of routing optimisation software with geographic information system (GIS) element in the case of waste collection for the reduction of operation costs and pollutant emissions. In Bing et al. (2014), the plastic waste collection is modelled as VRP and a tabu search algorithm is proposed to solve the problem. The objective is to minimise the total cost that comprises of transportation cost, labour cost and emission cost.

(c) End-of-life goods collection Ramos, Gomes, and Barbosa-Póvoa (2014a) modelled the recyclable packaging waste collection problem as a multi-product, multi-depot vehicle routing problem (MP-MDVRP) with two objective functions: distance and CO_2 emissions minimisation. A decomposition method is developed to solve the problem. In the same year, the same authors produced another paper which studied the multi-objective, multi-depot periodic vehicle routing problem with inter-depot routes (MDPVRPI) based on a set partitioning approach to solve a similar case study of recyclable packaging waste collection system. The idea is to support tactical and operational planning decisions of reverse logistics systems while considering economic (i.e. minimising variable costs), environmental (i.e. minimising greenhouse gas emissions), and social (i.e. balancing driver's schedule) objectives (Ramos, Gomes, and Barbosa-Póvoa, 2014b).

(d) Simultaneous distribution and collection Lin et al. (2014a) studied the Capacitated Vehicle Routing Problem with a Simultaneous Delivery and Pickup (VRPSDP) model in the case of delivering and collecting plastic carboys to minimise both economic and environmental costs. Other works that employed the VRP with simultaneous pickup and delivery (VRPSPD) can be found in Çatay (2010) and Dethloff (2001) but they did not clearly mention the importance to conserve the environment.

A relevant study by Salhi, Wassan, and Hajarat (2013) introduced the fleet size and mix VRP with backhauling (FSMVRPB) to better address the routing and distribution problem with a wide applicability for logistic companies that wish to determine the composition of their vehicle fleet as well as operating their delivery routes efficiently. Environmental factors could be incorporated into the model.

2.4 Multi-objective optimisation

Routing problems are regularly used to cater for real-life cases or applications where they are often set up with a single objective of minimising the costs. Yet the majority of the problems faced in industry, particularly in transportation and logistics are multi-objective in nature. There could be a number of costs related to a single tour in real life. Moreover, the objective is not necessarily the cost of the solution; other aspects could be added such as balancing the workload in terms of time, distance, and etc.

In general, multi-objective optimisation (MOO) can be defined as the practice of simultaneously solving at least two conflicting objectives in order to obtain optimal solution subject to a set of constraints. Thus, it can be stated in the following form (Demir, Bektaş, and Laporte, 2014b):

$$\begin{aligned} \underset{x}{\text{minimise}} \quad & f_1(x), f_2(x), \ldots, f_k(x) \\ \text{subject to} \quad & x \in S \end{aligned} \tag{2.7}$$

where $k \geq 2$ is the number of objective functions that need to be minimised simultaneously.

With conflicting objectives, sometimes a single solution that simultaneously optimises every objective cannot be achieved. An attempt to enhance an objective will result in losing the quality of the other objectives. A solution known as a non-dominated Pareto optimal is usually adopted. This is when the solution cannot be removed from consideration by replacing it with another solution which improves upon one of the objectives without worsening another. Therefore, the aim of formulating and solving an MOO problem would be finding such non-dominated solutions as well as quantifying the trade-offs in fulfilling the different objectives. In the multi-objective problem, the concept of optimality is changed since its purpose seeks good compromises or trade-offs rather than a single solution as in global search.

In the field of VRP, the use of multi-objective optimisation attracts more research attention since it offers new opportunities in defining the problems. The article by Jozefowiez, Semet, and Talbi (2008) provides a review of existing research related to multi-objective optimisation in routing problems. In their survey, routing problems are examined with regard to their definitions, their objectives and the

multi-objective algorithms proposed for solving them.

Ghoseiri and Ghannadpour (2010) studied the multi-objective VRPTW wherein the total required fleet size and total distance travelled are simultaneously minimised given the restriction on vehicle capacity and time windows. The problem is formulated using goal programming and further adapted genetic algorithm for efficiently solve it. The proposed genetic algorithm heuristics incorporate local exploitation in the evolutionary search and the concept of Pareto optimality for the multi-objective optimisation.

Ghannadpour et al. (2014) presented the study of the dynamic VRP with the concept of fuzzy time windows (DVRPFTW) with multi-objective. A direct interpretation of DVRPFTW is used as a multi-objective problem to the extent of minimising the total required fleet size, overall total travelling distance and waiting time imposed on vehicles as well as maximising the overall customers' preferences for the service.

There are many ways to tackle the multi-objective optimisation (MOO) problem. The classification of MOO methods provided by Demir, Bektaş, and Laporte (2014b) is summarised as below;

1. Generating methods – generating one or more Pareto efficient point(s) without any prior information
    (a) No-preference methods do not require any primary input and often simply return one Pareto optimal point. E.g. global criterion, multi-objective bundle method
    (b) Posteriori methods using scalarization approach convert the problem into single objective problem by real-valued objective function. E.g. weighting method, ϵ-constraint method
    (c) Posteriori methods using multi-objective approach use a multi-objective rank trial solutions based on the objective function values. E.g. non-dominating sorting genetic algorithm, ant colony optimisation
2. Preference-based methods – need additional input from decision maker
    (a) Priori methods – decision maker's preferences are sought and then included in initial formulation of single objective problem. E.g. values function method, lexicographic ordering, goal programming

(b) Interactive methods require the interaction with decision maker during solution process. E.g. interactive surrogate worth trade-off method, NIMBUS method

Based on the above categories, we have an interest and intend to pursue the posteriori methods with scalarization approach in formulating our vehicle routing model. The reasons are that the method seems to have less complexity in scalarizing the objective function and easy to implement in such a way that we do not have to wait to receive any information from decision maker beforehand.

The weighting method specifically, assigned each objective function with a non-negative weighting coefficient α_i for all $i = 1, 2, \ldots, k$ to minimise the weighted sum of the objectives. Through this, several objective functions are converted to single objective function. Then the problem can be solved with existing methods used to solve single-objective optimisation problem. The problem to be solved takes the form;

$$\begin{aligned} & minimize \sum_{i=1}^{k} \alpha_i f_i(x) \\ subject\,to \quad & x \in S \end{aligned}$$

where the weights are normalised in such a way that $\sum_{i=1}^{k} \alpha_i = 1$.

Among the existing works done using the posteriori methods with scalarization approach is Demir, Bektaş, and Laporte (2014b) which concentrated on four posteriori methods with scalarized objective functions. These are the weighting method, the weighting method with normalization, ϵ-constraint method, and the hybrid method that combined the adaptive weighting and ϵ-constraint methods. A study done by Soysal, Bloemhof-Ruwaard, and van der Vorst (2014) chose the ϵ-constraint method to solve multi-objective linear programming model for the case of a generic beef logistics network problem.

In this research, the weighting method is adapted to formulate a bi-objective problem in integrating the economic and environmental aspects. In other words, all the objectives to be optimised are included in a single function using a combination of mathematical operations. Specifically, the objectives are to minimise the total distance travelled and the amount of fuel consumed. Weights are assigned in the increment of 0.1 from 0.6 to 1.0. This aspect will be visited in the next chapter.

2.5 Fuel consumption and emission estimations

The importance of reducing the harmful effects of transportation on the environment has call for a better operational level planning in estimating and minimizing the vehicle emission. Thus, a good estimation model has to be well chosen and adapted in the route planning methods based on the type and nature of the problems. Demir, Bektaş, and Laporte (2011) present a comparative analysis of several emission models and assess their positive and negative characteristics.

Numerous energy and emission models are available and differ from their approaches and structures of their modelling as well as the data required for their computation. In this research, we emphasise on how to estimate the amount of fuel consumed by the vehicle. As we stressed the relationship of fuel consumption and the CO_2 emission in the previous chapter, we feel that the fuel consumption measure can be complex.

In Demir, Bektaş, and Laporte (2014a), the fuel consumption models are categorised into three main groups of increasing complexity levels: factor models, macroscopic models and microscopic models. Among all of the emission models described, we briefly discuss two that were utilised for this research. These are the methodology for calculating transport emissions and the energy consumption (MEET) model and the comprehensive modal emission modeling (CMEM). In the MEET model, an average speed is used to estimate the emissions for a variety of trips. The CMEM estimates the fuel consumption and emission rates with the instantaneous vehicle kinematic parameters, such as speed and acceleration to predict traffic emissions more accurately.

2.5.1 The methodology for calculating transport emissions and energy consumption (MEET) model

Hickman et al. (1999) published a report known as the methodology for calculating transport emissions and energy consumption (MEET) project funded by the European Commission. The model contains an assortment of approximating functions that are largely dependent on speed v and a number of fixed and predefined parameters for different groups of vehicles weighing from 3.5 to 32

tonnes. For smaller vehicles weighing less than 3.5 tonnes, the CO_2 rate of emissions is estimated by using this form of speed-dependency regression function $\epsilon = 0.0617v^2 - 7.8227v + 429.51$. For other vehicle classes, it is suggested by MEET to use the CO_2 emission factors (g/km) in the form of

$$\epsilon = K + av + bv^2 + cv^3 + d/v + e/v^2 + f/v^3$$

for an unloaded goods vehicles on a zero gradient road where K is a constant and a to f are predefined coefficients provided in this model.

This emission factors are tested with the standard condition, i.e. zero road gradient and empty vehicles, and are assessed as a function of average vehicle speed ranging from 10 to 90 km/h. Therefore, a few correction functions may be required for the effects of road gradient and vehicle load be significant on emissions estimation. Depending on the vehicle categories, the road gradient corrector function is in the form $GC = A_6v^6 + A_5v^5 + A_4v^4 + A_3v^3 + A_2v^2 + A_1v + A_0$ where A_0 to A_6 are constants. The function of taking load factor into account is as follows $LC = K + n\gamma + p\gamma^2 + q\gamma^3 + rv + sv^2 + tv^3 + u/v$ where γ is the load factors per gradient in percent, K is a constant and n to u are the coefficients.

Then according to MEET, the evaluation of CO_2 emission (g) can be calculated as;

$$E(v, D) = \epsilon \cdot GC \cdot LC \cdot Distance$$

2.5.2 The comprehensive modal emission modeling (CMEM)

The CMEM described in the report by Barth, Younglove, and Scora (2005) is developed at the University of California, Riverside to especially cater for heavy-duty diesel vehicle to measure instantaneous (i.e., modal) emissions and fuel consumption. This model requires detailed vehicle specific parameters for the estimations and is composed of three modules that are engine power, engine speed and fuel rate.

Demir, Bektaş, and Laporte (2014b) simplify the presentation of the model and expressed the fuel consumption of a vehicle on arc (i, j) with length d as a function

of speed v and load L as follows;

$$E(v_{ij}, L) = \lambda(kNV + w\gamma\alpha_{ij}v_{ij} + \gamma\alpha_{ij}L_{ij}v_{ij} + \beta\gamma v_{ij}^3)d_{ij}/v_{ij} \tag{2.8}$$

Let the parameter M be the total vehicle weight (kilogram) where $M = w + L$, w is the kerb weight (i.e., the weight of an empty vehicle) and L is the vehicle load.

Let $\lambda = \xi/\kappa\psi$ and $\gamma = 1/1000\eta_{tf}\eta$ be constants where ξ is the fuel-to-air mass ratio, κ and η are constants, ψ is the conversion factor of fuel from gram/second to liter/second, and η_{tf} is the vehicle drive train efficiency. The parameter k is the engine friction factor, N is the engine speed, and V is the engine displacement.

Furthermore, let $\alpha = \tau + g\sin\theta + gC_r\cos\theta$ be a vehicle-arc specific constant and let $\beta = 0.5C_d\rho A$ be a vehicle-specific constant where τ is the acceleration (meter/square second), g is the gravitational constant, θ is the road angle, ρ is the air density, A is the frontal surface area of the vehicle, and C_r and C_d are the coefficients of the rolling resistance and aerodynamic drag, respectively.

2.6 Summary

In this chapter, we presented an overview of the VRP including some of its variants and basic optimisation model. Furthermore, we reviewed other class of VRP which is the green VRP as this study also concerns about sustainable route planning. Among the works of green VRP, we described briefly the ones about the so-called pollution routing problem and the VRP with reverse logistic. We also provided a brief description of the multi-objective optimisation method. This method is adopted to formulate objective functions used in this study and this is presented in the next chapter. In addition, we also discuss some models which are known for assessing fuel consumption and vehicle emission. This model is adapted to estimate the amount of fuel consumed for the route which is used as a basis when we introduce the load factor into the formulation. In the next chapter, we discuss the savings method and present new formulations embedded with the load factor.

Chapter 3

The incorporation of the load factor in classical constructive heuristics

3.1 Introduction

In this chapter, we first briefly describe a classical constructive heuristic namely, the Clarke and Wright savings and then present two new formulations to incorporate the environmental effect. These adaptations could also be pursued in other constructive heuristics. We also provide the description of the fuel consumption function which is used to measure the CO_2 emission as well as the initial settings and adjustments made for computational experiments. We focus on investigating the load factor which is one of the main influences of the fuel consumption measure. The use of route direction is also explored. Computational results using the well known VRP instances are analysed and discussed.

3.2 Constructive methods

One of the methods used to generate an initial solution is to construct the solution starting from scratch. During the build-up of the solution, certain selection strategies are applied which can contribute to the final solution quality. For instance, for the TSP one way is to start from a random node, then add the nearest to it until the complete tour is formed. This greedy approach is usually called 'nearest neighbour rule'. We will focus on the savings method in this chapter,

though others such as the nearest neighbour search, the insertion-based and the sweep methods can also be attempted, see Laporte and Semet (2002).

3.2.1 The Clarke and Wright savings

The Clarke and Wright savings is one of the earliest and most studied methods for solving the capacitated vehicle routing problem. The basic version of this method is when two routes are originally performed by two vehicles of the same vehicle type (same capacity). These can be combined so that only one vehicle will carry out the task and thus yields the savings in the total cost. The savings S_{ij} of supplying customers on vertex i and vertex j in one route as opposed to supplying them individually directly from the depot can be calculated as follows;

$$\begin{aligned} S_{ij} &= 2d_{0i} + 2d_{0j} - (d_{0i} + d_{ij} + d_{0j}) \\ &= d_{0i} + d_{0j} - d_{ij} \end{aligned} \tag{3.1}$$

where d_{ij} denotes the cost or distance between customer i and customer j and vertex 0 is the depot.

Figure 3.1 illustrates the situation where the initial routes of serving the individual customers from the depot are represented by solid arrow and then the merged route is formed to obtain the savings in mileage. This is represented by the dashed arrows.

Larger values of S_{ij} imply that it is relatively attractive, pertaining to costs, to travel to vertex i and j on the same route such that customer at vertex j is visited immediately after the one at vertex i.

There are two versions of the savings algorithm, namely; the sequential and the parallel versions. In the sequential method, only one route is developed at a time until it is full (i.e. the vehicle capacity is violated), while in the parallel method several routes may be developed simultaneously. In brief, the first step of the algorithm is to calculate the savings for all combinations between two customers and then arrange them in descending order. The route is built by placing the combination of customers with the highest savings first and followed by the next largest as long as the vehicle capacity is not exceeded. The process is repeated until all customers are allocated in a route. In this study, the sequential version of the savings algorithm is used to generate the initial solution.

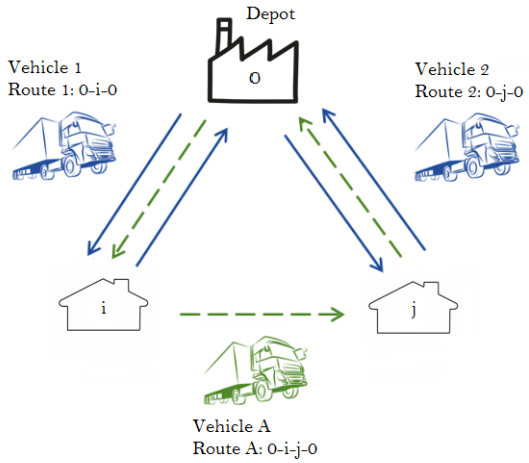

FIGURE 3.1: An example of the savings concept in a routing problem (same vehicle types)

Based on the Clarke and Wright savings formula, we have made some adjustment to its formulation (3.1) to incorporate the environmental factor as the original savings only involves the travel cost or simply, the economic factor. The environmental criteria that we include in the formulation is defined as the function of fuel consumed by a vehicle. This is directly related to the emission of CO_2 that poses a potential threat to the environment through climate change. Two ways on how the incorporation of environmental and economic factors are integrated in the savings formulation (3.1) are proposed. We refer to these two variants as (i) the integrated savings and (ii) the combined savings.

3.2.2 Formulation 1: The integrated savings

The incorporation of economic and environmental issues is reflected in the new environmental savings by defining the cost or distance between customer i and customer j as dependent on an environmental cost function. Factors that affect the fuel consumption and CO_2 emissions such as travelling distance d_{ij}, travelling times t_{ij}, vehicle load L_{ij}, speed v_{ij}, and road gradient θ_{ij} are embedded in the environmental cost function E_{ij}.

$$d_{ij} \rightarrow E_{ij} = f(d_{ij}, t_{ij}, L_{ij}, v_{ij}, \theta_{ij}, \dots)$$

Specifically in this research, E_{ij} denotes the function of the fuel consumption

which can be adapted from the emission models (see Demir, Bektaş, and Laporte (2011) and Demir, Bektaş, and Laporte (2014a)). The arguments included in the function depend on the model used to represent E_{ij}.

In the formulation called the integrated savings, the weight of arc (i, j) in the new savings algorithm $\bar{\tilde{S}}_{ij}$ is transformed in a way that both elements of economic and environmental criteria are included, see (3.2). In (3.2b), any other vertices in the road graph are denoted by k and l to differentiate the vertex i and vertex j with the vertices that give the maximum value of distance, d_{kl} or fuel consumption E_{kl}.

$$\bar{\tilde{S}}_{ij} = \tilde{C}_{0i} + \tilde{C}_{0j} - \tilde{C}_{ij}, \tag{3.2}$$

where in each arc, the cost is defined as

$$\tilde{C}_{ij} = \alpha \hat{d}_{ij} + (1 - \alpha)\hat{E}_{ij} \tag{3.2a}$$

$$\hat{d}_{ij} = \frac{d_{ij}}{\max\limits_{\forall k,l} |d_{kl}|}, \hat{E}_{ij} = \frac{E_{ij}}{\max\limits_{\forall k,l} |E_{kl}|}, \alpha \in (0, 1). \tag{3.2b}$$

Note that both d_{ij} and E_{ij} are normalised. For illustration $\tilde{C}_{ij}$ is shown in Figure 3.2.

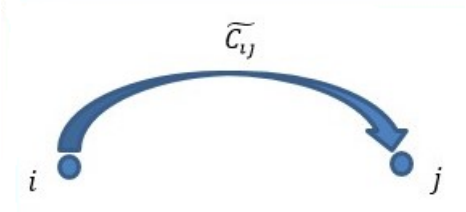

FIGURE 3.2: Arc (i, j) based on formulation 1

3.2.3 Formulation 2: The combined savings

In this formulation, we combine two different savings between customer i and j with a weight attached. The first one is the regular savings S_{ij} calculated based on the length between customers (3.1). The second one is the new environmental savings, which is based on the fuel consumption. The environmental savings which

we define by S_{ij}^E is computed in the following form similar to (3.1);

$$\begin{aligned} S_{ij}^E &= 2E_{0i} + 2E_{0j} - (E_{0i} + E_{ij} + E_{0j}) \\ &= E_{0i} + E_{0j} - E_{ij}, \end{aligned}$$

where E_{ij} is the function of the fuel consumption adapted from the emission models.

Subsequently, both cost and environmental savings formulae are combined to obtain the overall savings $\tilde{S_{ij}}$ as shown in (3.3). Note that, the indices k and l below are the vertices in the road network other than i and j that gives the maximum value of the regular savings S_{kl} or the environmental one S_{kl}^E.

$$\tilde{S_{ij}} = \alpha \hat{S_{ij}} + (1 - \alpha)\hat{S_{ij}^E}, \tag{3.3}$$

where $\hat{S_{ij}} = \frac{S_{ij}}{\max\limits_{\forall k,l} |S_{kl}|}$ and $\hat{S_{ij}^E} = \frac{S_{ij}^E}{\max\limits_{\forall k,l} |S_{kl}^E|}$ are normalised and $\alpha \in (0, 1)$.

It is observed that the value of $\hat{S_{ij}^E}$ must be significant for the whole savings to be worthwhile. The constant parameter α (or weight) in the overall savings reflects the impact of economic or environmental aspects on the overall savings when solving the green VRP. For instance when $\alpha = 1$, the problem reduces to the classical VRP where the objective is the operational cost minimisation only representing the economic element as usually done. On the other hand, when $\alpha = 0$ the environmental component is now the only criteria considered with the emphasis on the reduction in fuel consumption and CO_2 emission. A representation of S_{ij} is shown in Figure 3.3.

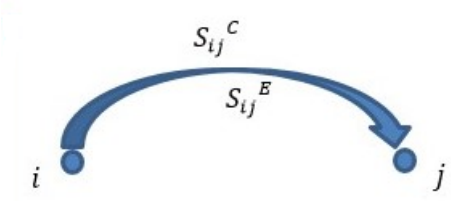

FIGURE 3.3: Arc (i, j) based on formulation 2

3.2.4 The fuel consumption function

Some of the existing emission models that are used to evaluate the energy and fuel consumption can be found in Demir, Bektaş, and Laporte (2011) and Demir, Bektaş, and Laporte (2014a). We introduced a new variable E_{ij} which can be extracted from any of those models to represent the environmental criteria. Among those models, we investigate two known models, which are the MEET and the CMEM models to measure E_{ij}.

The load factor

The vehicle load is one of the most important factor that affects the routing in terms of the fuel consumption of the vehicle and subsequently contributes to the CO_2 emission. We focus our investigation on the CMEM model as it is shown to be more thorough and widely used in estimating the CO_2 emission. Other models could also be worth exploring in the future for further comparison.

Initially, we adapted this model to observe the single effect of changing load to fuel consumed by the vehicles. Thus, we try to simplify the function by using the average speed within the specified network. The assumption is that the vehicle is travelling throughout the network with a constant speed so that we can observe how the varying load will affect the routing. This problem could be expanded with the use of predefined vehicle speeds for a particular arc (i, j) which represent different types of road. For example, when the vehicle takes a trip on urban roads, the recommended speed $v = 30$ km/h, for outside-urban roads $v = 50$ km/h and on motorways $v = 70$ km/h, etc.

To simplify the notation in Equation (2.8), let $\alpha_0 = \lambda(kNVv_{ij}^{-1} + w\gamma\alpha_{ij} + \beta\gamma v_{ij}^2)$ and $\alpha_1 = \lambda\gamma\alpha_{ij}$ be constants. The function then will take the form as follows;

$$\begin{aligned} E_{ij} &= \lambda(kNVv_{ij}^{-1} + w\gamma\alpha_{ij} + \beta\gamma v_{ij}^2)d_{ij} + \lambda\gamma\alpha_{ij}L_{ij}d_{ij} \\ E_{ij}^p &= (\alpha_0^p + \alpha_1^p L_{ij})d_{ij} \end{aligned} \tag{3.4}$$

The parameter p is the pth type of vehicle ($p = 1, \ldots, P$) with P representing the number of vehicle types. If an homogenous fleet is used (i.e. $p = 1$), we can ignore p from the notation. A larger vehicle (heavy-duty vehicle) will have a large frontal surface area and therefore will require a huge amount of aerodynamic drag force which influences the power demand, fuel economy and emissions. Using the

parameter values adapted from Demir, Bektaş, and Laporte (2014b), where the vehicle is traversed at the constant speed of 55 km/h (see Table 3.1) the function E_{ij} will reduce to

$$E_{ij} = [(1.52933 \times 10^{-4}) + (8.40323 \times 10^{-9})L_{ij}]d_{ij}$$

TABLE 3.1: Parameters used in the fuel consumption function for computational tests (Demir, Bektaş, and Laporte, 2014b)

Notation	Description	Typical values
w	Kerb weight (kg)	6350
ξ	Fuel-to-air mass ratio	1
ψ	Conversion factor of fuel (g/s to l/s)	737
κ	Heating value of a typical diesel fuel (kJ/g)	44
η	Efficiency parameter for diesel engines	0.9
η_{tf}	Vehicle drive train efficiency	0.4
k	Engine friction factor (kJ/rev/l)	0.2
N	Engine speed (rev/s)	33
V	Engine displacement (l)	5
g	Gravitational constant (m/s^2)	9.81
C_r	Coefficient of the rolling resistance	0.01
C_d	Coefficient of the aerodynamic drag	0.7
ρ	Air density (kg/m^2)	1.2041
A	Vehicle frontal surface area (m^2)	3.912

Observation

Because of the load factor in E_{ij}, it becomes more difficult to compute the environmental savings. Note that the normal savings are usually symmetrical. However, when we consider the decreasing load when making delivery (and the increasing load when they are being collected), the vehicle load of travelling from i to j is not the same as the journey from j to i. Hence, the introduction of the load factor renders the environmental savings assymetrical. This makes the computation and the choices of routes taken for the delivery vehicle routing problem very different to its counterpart the collection vehicle routing problem.

From now on, the simplified formula of the model as in formulation (3.4) will be used as the measure to calculate the environmental criterion. The measure will compute the fuel consumption based on the average speed with the effect of the varying load when the vehicle travels from one node to another. This measure is also used as a base to modify some of the constructive heuristics and the commonly

used local search operators for investigating the effect of the vehicle load in the routing. The latter will be explored in the next two chapters.

Adjusting the vehicle capacity and customers' demand

In this study, note that the parameter values in the model are adapted from Demir, Bektaş, and Laporte (2014b) and the test instances used in their work are generated from a real case study. Whereas the test instances used in this study are the VRP data sets from the literature with the smallest and the largest vehicle capacity restriction of 140 and 30000, respectively.

By simple mathematical manipulation, the fuel consumption function as in (3.4) is separated into two components of α_0 and α_1. We observed that if the vehicle load (demand) L_{ij} value is way too smaller than the vehicle kerb weight w, we can hardly see any effect of varying load in the routing process. This is because of the constant parameters in the first component, α_0 such as the kerb weight, among others, dominated the latter component α_1. The second component α_1 depends on the vehicle load L_{ij} which is determined by the customer's demand q_i and restricted by the vehicle capacity Q. Hence, it is important for us to come out with a solution to have a meaningful fuel consumption measure with the monitoring of the load factor while using the same parameter values and test instances.

To counter this issue, we adjusted the vehicle capacity in the instance data sets so that it will be equal to the empty vehicle weight that we use in the model. In this sense, our approach follows that of Bektaş and Laporte (2011) who conclude that vehicles transport around as much freight as their kerb weight. This conclusion was based on their review where they discovered that the ratio of the vehicle kerb weight to its Gross Vehicle Weight Rating (GVWR) is around 50% for several vehicle types.

If the adjusted vehicle capacity $Q' \equiv w$, then we have the adjusted constant, $k = w/Q \cdot \delta$ where δ is the correction factor coefficient. As a result, the adjusted vehicle capacity is composed as $Q' = kQ$ and the adjusted customers' demand as $q'_i = kq_i$.

Adjusting the savings algorithm for the effect of changing load

An illustration of how the vehicle load is changing in an example of the savings situation for the delivery problem is given in Figure 3.4. This relates to the adjustment of the savings algorithm to take on the load factor in the problem. Thus, two new savings formulations are adjusted as follows. For the integrated savings formulation (3.2), the integrated cost $\tilde{C}_{ij}$ is modified so that we can address the changing load that occurs along the route. Thus, the adjusted formula for the integrated savings (3.5) is as follows;

$$\begin{aligned} \bar{\tilde{S}}_{ij} &= \tilde{C}_{0i} + \tilde{C}_{i0} + \tilde{C}_{0j} + \tilde{C}_{j0} - (\bar{\tilde{C}}_{0i} + \bar{\tilde{C}}_{ij} + \bar{\tilde{C}}_{j0}) \\ &= \tilde{C}_{0i} + \tilde{C}_{i0} + \tilde{C}_{0j} - (\bar{\tilde{C}}_{0i} + \bar{\tilde{C}}_{ij}) \end{aligned} \tag{3.5}$$

where $\tilde{C}_{ij} = \alpha\hat{d}_{ij} + (1-\alpha)\hat{E}_{ij}$ and $\bar{\tilde{C}}_{ij} = \alpha\hat{d}_{ij} + (1-\alpha)\hat{\bar{E}}_{ij}$.

We also modify the combined savings formulation (3.3) as we have to take into consideration the vehicle load while the vehicle traverses along the arc (i, j).

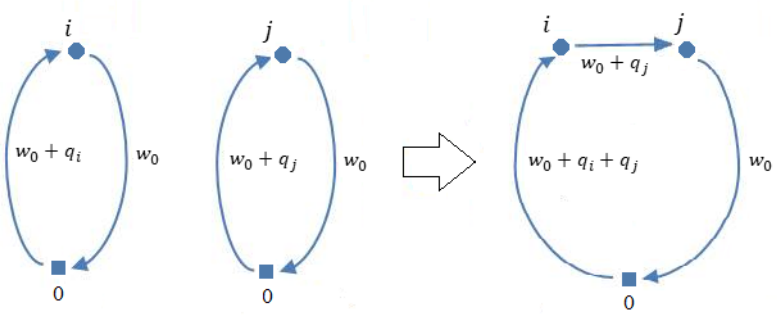

FIGURE 3.4: Vehicle load consists of vehicle weight w and customers' demand q from i to j for delivery problem

The adjusted formula for environmental savings of the route from i to j (3.6) is as follows;

$$\begin{aligned} \tilde{S}^{E}_{ij} &= E_{0i} + E_{i0} + E_{0j} + E_{j0} - (\tilde{E}_{0i} + \tilde{E}_{ij} + \tilde{E}_{j0}) \\ &= E_{0i} + E_{i0} + E_{0j} - (\tilde{E}_{0i} + \tilde{E}_{ij}) \end{aligned} \tag{3.6}$$

$\tilde{E}_{0i}$ is the fuel consumption function for arc (i, j) after i and j are combined in the

same route. Note that when back to the depot, E_{j0} and $\tilde{E_{j0}}$ have the same value since the vehicle returns empty by carrying the same load (its weight only).

3.3 Computational experiments

The formulae are coded in Visual Studio C++ and executed on an Intel Core i5-3230M CPU at 2.6GHz PC with 6GB RAM. Computational tests were undertaken on a sample of 46 VRP benchmark test problems from the literature. These problems are categorised into three sets according to the size of customers which we refer to as small, medium and large data sets. For ease, the test problems are labelled with the initial of the first author from where the problems are first presented in. Note that, some VCAP are in hundreds and thousands. The extreme one, F2 also has a large amount of customer's demand. Details of the data sets are given in Table 3.2. In this study throughout all the chapters, we carry out all the experiments on these 46 test problems using the same software and computer.

In the small data set, we utilised 14 problems ranging in size from 21 to 199 customers. The first three problems are from Fisher (1994) which are extracted based on the real-life application. Problem F1 and F3 are concerned with the groceries deliveries, while problem F2 is concerned with the delivery of vehicle parts to gasoline service stations. Problems C1 to C7 are from Christofides and Eilon (1969). Problems C4 to C7 are created by generating a random customers' location from a uniform distribution. Problems C8 to C11 are from Christofides, Mingozzi, and Toth (1979). Problem C8 was composed by combining the customers in problems C5 and C7 with the depot and vehicle capacities as in problem C7. Problem C9 was composed by adding the first 49 customers from problem C6 to problem C8. Customer locations are clustered in problems C10 and C11 to resemble problems to those observed in practice.

The medium data set is from Golden et al. (1998) which have 20 test problems with the number of customers varying between 200 and 483. The first eight instances have restriction on route-length. The customers in problems G1 to G8 are positioned in concentric circles around the depot. The location of customers in problems G9 to G12 is set in concentric squares with the depot at one corner. In problems G13 to G18, the customers are located in concentric squares around the depot.

TABLE 3.2: Data sets

Small			Medium			Large		
# (n)	VCAP	RL	# (n)	VCAP	RL	# (n)	VCAP	RL
F1 (44)	2010	-	G1 (240)	550	650	L1 (560)	1200	1800
F2 (71)	30000	-	G2 (320)	700	900	L2 (600)	900	1000
F3 (134)	2210	-	G3 (400)	900	1200	L3 (640)	1400	2200
			G4 (480)	1000	1600	L4 (720)	1500	2400
C1 (21)	6000	-	G5 (200)	900	1800	L5 (760)	900	900
C2 (22)	4500	-	G6 (280)	900	1500	L6 (800)	1700	2500
C3 (29)	4500	-	G7 (360)	900	1300	L7 (840)	900	900
C4 (32)	8000	-	G8 (440)	900	1200	L8 (880)	1800	2800
C5 (50)	160	-	G9 (255)	1000	-	L9 (960)	2000	3000
C6 (75)	140	-	G10 (323)	1000	-	L10 (1040)	2100	3200
C7 (100)	200	-	G11 (399)	1000	-	L11 (1120)	2300	3500
C8 (150)	200	-	G12 (483)	1000	-	L12 (1200)	2500	3600
C9 (199)	200	-	G13 (252)	1000	-			
C10 (120)	200	-	G14 (320)	1000	-			
C11 (100)	200	-	G15 (396)	1000	-			
			G16 (480)	1000	-			
			G17 (240)	200	-			
			G18 (300)	200	-			
			G19 (360)	200	-			
			G20 (420)	200	-			

Test problem number; n Number of customers;
VCAP Vehicle capacity constraint; RL Route-length constraint

The large data set from Li, Golden, and Wasil (2005) consists of 12 problems with the number of customers ranging from 560 to 1200. All problems have a certain route length restriction. For simplicity, these data sets can also be downloaded from the website CHLO (2017).

3.3.1 Evaluation procedures for assessing the environmental impact

We discuss some of the evaluation procedures to assess the effect of load in the new formulations. We first describe how we set up the objective function, followed by the effect of the route direction on the solutions.

Evaluation of the bi-objective function

Here, our aim is to minimise the sum of a weighted bi-objective function. The weights, α or the decision maker's preference is set at a step of 0.1 from 0.6 to 1.0. In general, the aggregated objective function is calculated as $\alpha f_1 + (1-\alpha) f_2$. We consider the first function f_1 to reflect the monetary cost where it is composed of the travelling cost measured by the Euclidean distance, d_{ij}. The second function f_2 is considered as the social costs incurred by the choice of the decision maker by placing a certain weights to it. This is measured by the fuel consumption, E_{ij} to capture the CO_2 emission. Therefore, the objective function is given by

$$Z(x) = \alpha\mu_1 \sum d_{ij} + (1-\alpha)\mu_2 \sum E_{ij} \tag{3.7}$$

where μ_1 is the unit travel cost which is the fuel price per unit distance and μ_2 is the unit social CO_2 cost per litre of fuel.

The definition of the social cost of CO_2 is given by Price, Thornton, and Nelson (2007) as shown below;

> *The social cost of carbon measures the full global cost today of an incremental unit of carbon (or equivalent amount of other greenhouse gases) emitted now, summing the full global cost of the damage it imposes over the whole of its time in the atmosphere.*

According to them, the social cost of carbon is of importance as it indicates what society should, in theory, be willing to compensate now to prevent the imminent damage caused by incremental carbon emissions. For example, a social cost of, say, £20 per tonne of CO_2 means that releasing an extra tonne of CO_2 at the moment has the same influence on society's expected welfare by way of reducing a representative consumer's consumption by £20 today.

The value for μ_1 is set as £0.17, which is the average diesel fuel cost per km for 20 MPG vehicle. Based on 2007 social cost of CO_2 which is approximately £29/tonne, we convert this value to obtain μ_2 which is set to £0.7656 corresponding to the amount of CO_2 in diesel which is 2.64 kg/litre. These values can obviously change though the methodology remains unchanged.

Evaluation of the reverse direction of the route

Usually, the optimal solution obtained from the distance-based methods can be traversed in both ways without affecting the solution. For example, a case of visiting A, B, C from the depot 0 can be either (a) 0-A-B-C-0 or (b) 0-C-B-A-0, see Figure 3.5. Both (a) and (b) leading to the same total mileage cost (i.e. $C_d = 25$). For simplicity in this example, we use the environmental cost C_e as the total weighted distance wherein each arc (i,j) the distance is multiplied by the vehicle load. The values of C_e for both cases appeared differently and we found the percent deviation between the two is approximately 21%.

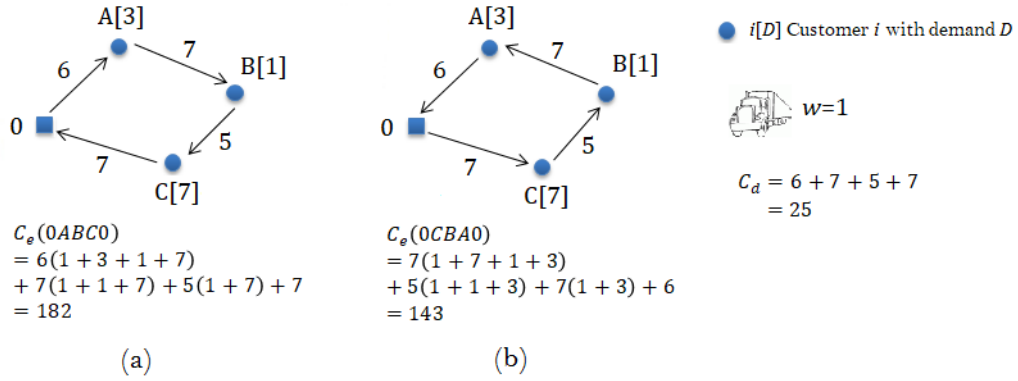

FIGURE 3.5: An example route of visiting A, B, and C from the depot 0

Regardless of the route directions, the route cost in terms of the distance is always the same. This is also true for both delivery and collection problems given that the distance matrix is symmetric. On the contrary, this is not the case in view of the environmental criteria such as the load factor in the problem. This is because each customer requires a different amount of goods to be delivered (collected) to (from) them. When considering the load that the vehicle carries while travelling from a customer to another in obtaining the solutions, due to the problem structure, the road network becomes directed. Thus, we have the asymmetrical fuel consumption cost. The decision models for delivery and collection cases are constructed differently. This is explained further in Kara, Kara, and Yetis (2007). In this study, we are adapting the load factor in the decision models focusing on the delivery case.

There are some cases or other VRP variants where the established routes cannot be reversed. This includes VRP with Pickup and Delivery for both mixed and simultaneous cases where the simple version of it considers the delivery is performed

before the pickup. This is also true for the Asymmetric VRP variant. Moreover, there is a case of transportation of hazardous materials where the routing depends on the vehicle load and the population exposed when an incident happens (see Bula et al. (2017)).

Effect of the reverse route

The resulted route which can be assessed from two directions gives the same distance length, but different in the amount of fuel consumed given that the demand of the customers in the route varies. When we apply the reverse route evaluation procedure, two scenarios of the solutions which we refer to as the best and the worst-case solutions are obtained. The best-case solution returns the set of routes with the route direction that gives the minimum fuel consumption while the worst-case is the set of routes with the other route direction. We can relate to the former as the optimistic scenario whereas the latter reflects the pessimistic case. If the load factor is not used, the decision maker could opt for either option without knowing.

Mathematically, the objective function of the solution of the best and the worst-case in general can be written as

$$Z_{best} = \sum_{k=1}^{NR} F_b(R_k) \qquad \text{and} \qquad Z_{worst} = \sum_{k=1}^{NR} F_w(R_k)$$

where $F_b(R_k)$ and $F_w(R_k)$ refer to the best and the worst fuel consumption for each route k.

Illustrative example

To illustrate the effect of the route direction, consider the following example consisting of two routes made up of three nodes each as highlighted in Figure 3.6. This is an extension to the problem in Figure 3.5. In this example, two extreme cases of solutions are shown as (a) the best-case and (b) the worst-case. The percentage deviation of C_e is found to be approximately 33% between these two cases while the length cost C_d remains unchanged. This is such an interesting discovery and that the best-case solution appears to be very attractive.

If the solution is chosen at random, there are more possibilities of how the solution can be. In this example, two additional combinations of routes can be obtained to form the solutions. The first one would be R_1 and R_2' and the other would be

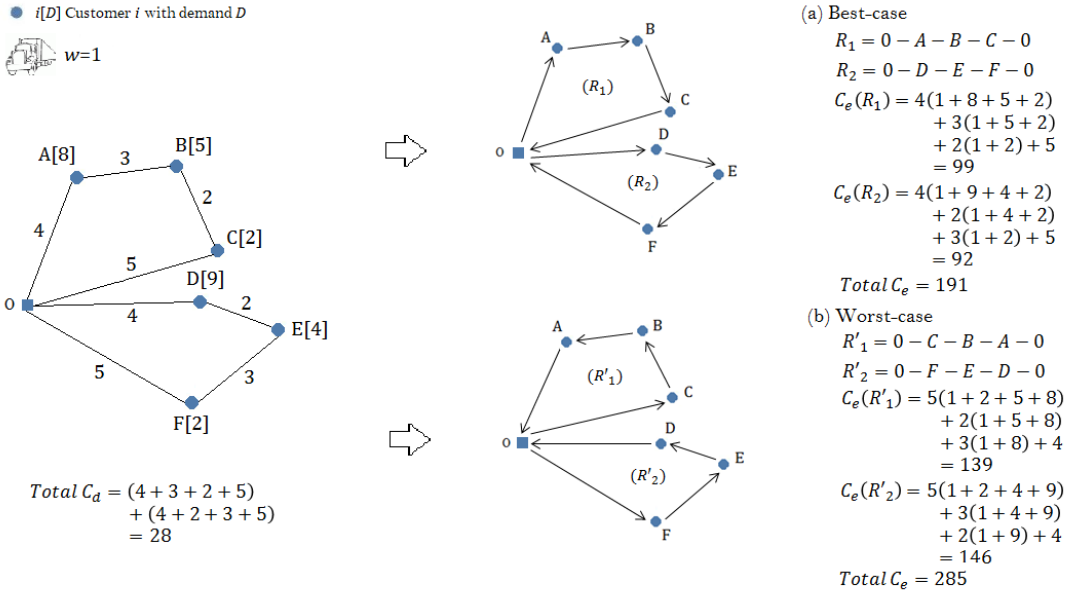

FIGURE 3.6: An example of routing to show the effect of the reverse route

the opposite, R'_1 and R_2. In Figure 3.7, we show the solutions for this example in terms of C_e in a range between the best and the worst-case. The solution at point (i) is acquired when R'_1 and R_2 formed the solution, and vice versa for point (ii).

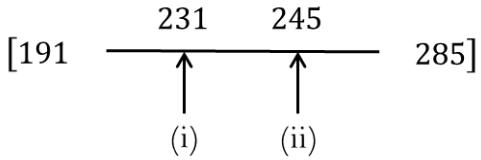

FIGURE 3.7: A range of solutions for example in Figure 3.6

Methodology

Without assessing both directions of routes while taking into account the load factor, the solution could be formed in 2^k possibilities. This includes all possibilities in between. In our case, we are showing only the two extreme cases; the best and the worst. This is depicted in Figure 3.8. C_1 denotes the best-case solution and C_2 is the nearest solution to it. Similar representation for C_{2^k} which denotes the worst-case solution with C_{2^k-1} being the nearest solution to it. The solution also could end up randomly in between the interval of $k = [2, 2^k - 1]$. For instance, in the previous example $k = 2$. When using the non-load-based methods, we want to assess both scenarios using the best and the worst routes direction. In this case,

the solution with the best-case scenario will give a chance for the non-load-based methods to be improved when the load factor is present. More importantly for the load-based methods, we select the best-case scenario as its solution at each time.

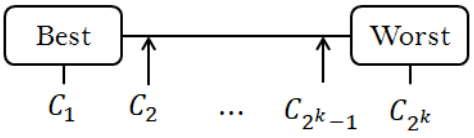

FIGURE 3.8: A range of solutions between two extreme cases

In Figure 3.9, we display how the solutions are recorded when performing the savings method. Those solutions are also set as our initial solution before carrying out the refinement methods (i.e. local searches) which will be presented in the next two chapters.

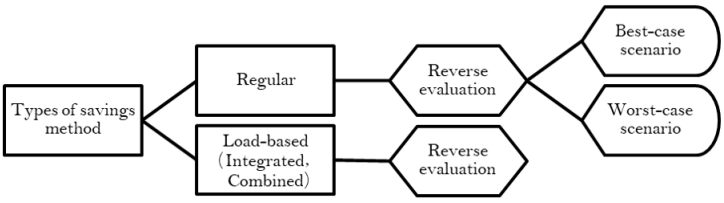

FIGURE 3.9: Way of recording the results of the savings method

Observation

To show the importance of evaluating the reverse direction of routes, we perform a simple analysis to our savings solutions comparing with and without the evaluation procedure. We calculate the percent deviation of the total objective function Z as $(Z_{wo} - Z_w)/Z_{wo} \times 100$ where the subscripts w and wo represent the solution with and without the evaluation of the reverse route direction respectively for each method. The positive percentage of deviation gives us an improvement after evaluating such a procedure.

We provide the sample result of the analysis in Figure 3.10 for regular, integrated and combined savings with $\alpha = 0.6, 0.7, 0.8$, and 0.9. We started with $\alpha = 0.6$ as one expects the balance to start from midway onwards. A step of 0.1 is used although some other values could also be used. Given the α values, almost all

problems benefit from this evaluation in which the solutions are improved. The highest average percent deviation reaches nearly 1% when using the combined savings in the small data set for $\alpha = 0.6$.

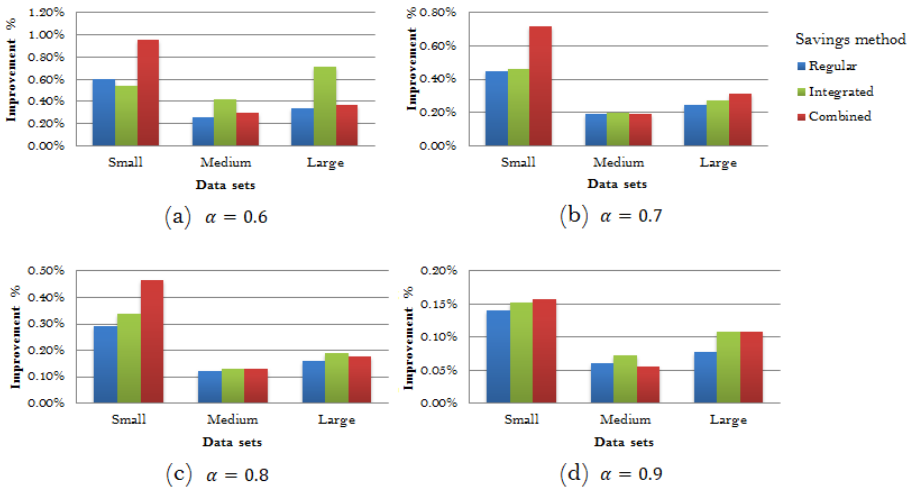

FIGURE 3.10: The average percentage of deviation after the evaluation of the reverse direction of routes for savings methods

We also illustrate the effect of evaluating the reverse direction of routes as in Figure 3.11. Referring to the figure, the trend of the average percentage deviation seems to be decreasing when α is increasing. This is somewhat obvious. This finding is consistent with the fact that the higher the value of α, the lower the value of the social cost of CO_2, i.e. the environmental criterion. Note that when $\alpha = 1$, there is no consideration of the environmental criterion. From here, we conclude that it is important to have the reverse direction of each route being assessed to gain some improvement when considering the load factor.

3.3.2 Computational results

To assess the behaviour of the proposed savings-based heuristics, we record the solutions over five values of α ranging from 0.6 to 1 with an increment of 0.1. We calculate the percentage deviation of the total objective function (3.7) between our method and the best-case for the regular savings. The mathematical expression to calculate percent deviation is:

$$Dev(\%) = |Z - Z_k|/Z \times 100 \tag{3.8}$$

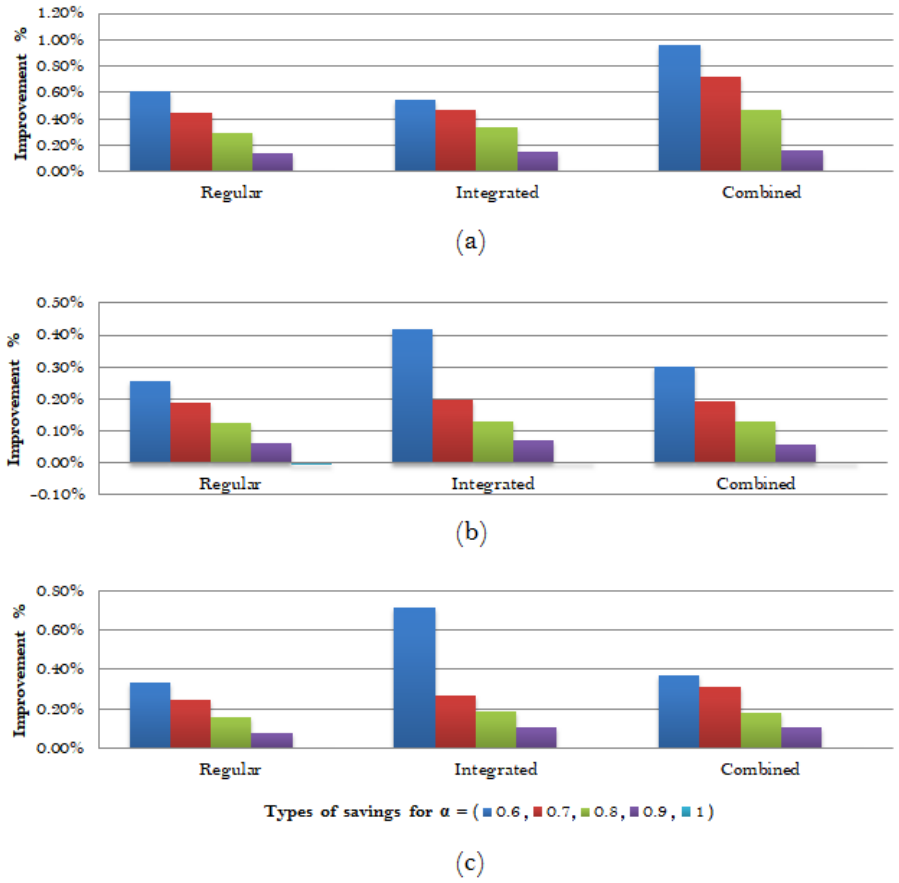

FIGURE 3.11: The average percentage of deviation after the evaluation of the reverse direction of routes for (a) small, (b) medium and (c) large data sets

where Z is the total objective function for regular savings method and Z_k for integrated and combined savings represented by subscript $k = i$ for the integrated savings and $k = c$ for the combined savings respectively. The positive value of a percentage deviation indicates that the solution of our modified method improved with the reduction of the total objective function from the regular savings method.

Integrated savings results

For the integrated savings method, we present the average of percent deviation for each small, medium and large data sets as in Table 3.3. The frequency of instances, excluding ties, yielding a positive and negative deviation percentage are recorded as #best and #worst, respectively. The highest and the lowest deviation percentage are recorded as Best % and Worst %, respectively.

For small data set, at $\alpha = 0.7$, the highest percentage of deviation is acquired at

TABLE 3.3: Summary table for integrated savings against the best case of the regular savings

Data set		α				
		0.6	0.7	0.8	0.9	1.0
Small	AvgDev(%)	-1.22	0.18	0.13	-0.26	0.00
(14 instances)	#best	1	7	8	6	0
	#worst	4	6	5	5	0
	Best(%)	0.66	**5.98**	5.80	4.54	0.00
	Worst(%)	**-13.45**	-8.15	-7.38	-5.18	0.00
Medium	AvgDev(%)	-0.79	-2.93	-1.41	-0.78	-0.01
(20 instances)	#best	5	3	6	6	3
	#worst	15	**17**	14	14	4
	Best(%)	2.60	1.34	3.64	1.76	0.98
	Worst(%)	-5.29	-7.86	-5.58	-3.70	-1.60
Large	AvgDev(%)	**0.52**	-1.06	0.39	0.49	-0.07
(12 instances)	#best	**8**	5	7	**8**	4
	#worst	4	7	5	4	5
	Best(%)	2.15	2.63	3.34	3.63	0.98
	Worst(%)	-0.74	-5.38	-4.36	-3.76	-1.12

5.98% whereas the lowest value could be found in the same data set at $\alpha = 0.6$ which is -13.45%. With the same α value, the average percent deviation is found to be at the highest in the large set, although a 0.52% improvement is rather small. Out of 12 instances, 8 or approximately 67% of instances, in the large data set for both $\alpha = 0.6$ and 0.9 produced positive improvement solutions. This result can be considered a success among all other α values and in other data sets. On the other hand, 17 of the 20, or 85% of instances in the medium data set on $\alpha = 0.7$ yielded solutions with a negative percent deviation when compared to solutions from the regular savings method.

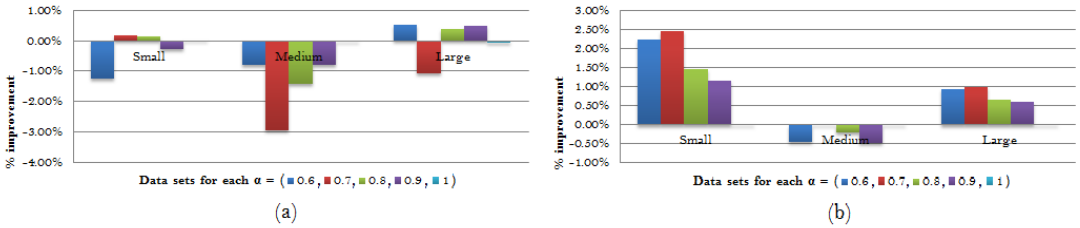

FIGURE 3.12: The average percentage of deviation of (a) the integrated savings and (b) the combined savings against the regular savings (best-case scenario)

Combined savings results

For the combined savings method, the small data set seems to produce much better quality solution than the other two data sets. This can be depicted in Table 3.4. The average percent deviation is found to be the highest at 2.46% when $\alpha = 0.7$. This corresponds to the highest frequency of positive percent deviation which records 11 of the 14 instances or around 79% of the instances occurred both when α is set to 0.6 and 0.7. The best percent deviation is also attained from the small data set at 8.64% when $\alpha = 0.9$. In spite of that, the worst percent deviation could be seen from the medium data set with -6.11% when $\alpha = 0.6$. Out of 20 instances, 12 of them gave negative percent deviation when the values of α are 0.6 and 0.8 resulting in a negative average percent deviation.

TABLE 3.4: Summary table for the combined savings against the best case of the regular savings

Data set		α				
		0.6	0.7	0.8	0.9	1.0
Small	AvgDev(%)	2.25	**2.46**	1.48	1.15	0.00
(14 instances)	#best	**11**	**11**	9	7	0
	#worst	3	3	4	4	0
	Best(%)	8.30	8.42	8.52	**8.64**	0.00
	Worst(%)	-2.60	-1.51	-1.50	-3.90	0.00
Medium	AvgDev(%)	-0.46	0.01	-0.20	-0.48	0.00
(20 instances)	#best	8	10	8	10	0
	#worst	**12**	10	**12**	10	0
	Best(%)	3.13	2.02	2.59	1.53	0.00
	Worst(%)	**-6.11**	-3.34	-3.35	-4.34	0.00
Large	AvgDev(%)	0.93	0.99	0.65	0.59	0.00
(12 instances)	#best	9	8	7	9	0
	#worst	3	4	5	3	0
	Best(%)	3.60	2.84	3.00	4.09	0.00
	Worst(%)	-2.51	-1.59	-2.97	-1.65	0.00

CPU times

In terms of the computing time, we recorded the time beginning from the calculation of the savings matrix until the final solution is found. The average total time when computed over 46 problems are 23, 34 and 30 milliseconds respectively for regular, integrated and combined savings as shown in Table 3.5. In comparison to the analysis made between the integrated and combined savings, we concluded

that the solutions obtained from the combined savings method are more appropriate in featuring the load factor in the initial stage. This is exemplified by generally having a more positive percent deviation as in Figure 3.12 as well as slightly less computing time than the integrated savings.

TABLE 3.5: Average computational time (seconds)

Data set	Savings		
	Regular	Integrated	Combined
Small	0.001	0.003	0.005
Medium	0.014	0.018	0.017
Large	0.064	0.095	0.081
Overall	0.023	0.034	0.030

3.4 Summary

In this chapter, we present the classical Clarke and Wright savings method as the constructive heuristic for the capacitated vehicle routing problem. To incorporate the environmental criteria, we introduced two new mathematical formulations of the savings namely, the integrated and the combined savings. These formulations can easily be adapted to other constructive heuristics such as the nearest neighbour or the insertion-based heuristics. In this study, fuel consumption is used to measure the CO_2 emission while concentrating on the load factor. As the new problem is assymetric, both directions of the established routes are evaluated to see the effect of the load in the amount of fuel consumption. This leads to two very different scenarios namely the best and the worst-case as the user, if not aware of the load factor, could choose whichever option including the worst one. This evaluation provides a range where the solution based on 'regular' measures could be. Given our experiments, in the following chapters, the combined savings method together with the regular savings will be used in generating the initial solution.

Chapter 4

Incorporation of a load factor in basic local search operators

4.1 Introduction

In this chapter, we first briefly describe the first three operators, which consists of (i) 2-opt, (ii) (1,0) relocate and (iii) (2,0) relocate. These operators are considered as basic since they are simple and fast. We introduce the load factor into the formulation of the operators and present the necessary mathematical formulations for each operator with the attached vehicle load factor. The newly developed operators are firstly shown with an illustrative example and followed by full computation results based on the known VRP instances. We also highlight the computational comparison between brute-force and the new formula-based search.

4.2 Design of the new operators

For clarity of presentation, in this chapter, we define the Euclidean distance or the length between vertex m and n as $d(m,n) = d_{mn} = \sqrt{(x_m - x_n)^2 + (y_m - y_n)^2}$. The demand of customer m is rewritten as $q(m) = q_m$. The k^{th} customer in route i is written as $R_i(k)$ and $nc(R_i)$ is the number of customers in route i.

In accommodating the load factor in the operators, the route cost is calculated in terms of the amount of fuel consumed by a vehicle when travelling along the route. The fuel consumption function adapted from (3.4) is also rewritten as $E(m,n) = E_{mn} = (\alpha_0 + \alpha_1 L(m,n))d(m,n)$. The parameter $L(m,n) = L_{mn}$ is the

vehicle load when travelling along the arc (m, n) and can be computed once the route is established. By accumulating the demand from the n^{th} to the $nc(R_i)^{th}$ customer in route i, this leads to $L(R_i(m), R_i(n))$ being defined as follows:

$$L(R_i(m), R_i(n)) = \sum_{a=n}^{nc(R_i)} q(R_i(a))$$

Strategy for selecting alternating solutions

In this study, we emphasise how the load factor can be incorporated in the routing problem by transforming and modifying the necessary formulation and procedures. Hence, we decide to carry out the best improvement strategy as this can lead to a good local optimum although the procedure may be longer. Methods that are known to be able to speed up the search process such as neighbourhood reduction and data structure can also be explored along with the best improvement strategy though they are not used in this research.

Effect of the reverse route

When the local search operator is applied, the procedure of evaluating the reverse route takes place as in Figure 4.1. Both the best and the worst-case scenarios are assessed when the solution is improved with the non-load-based which we refer to as the classical operator. More importantly, we opt for the best-case solution before commencing the load-based operator. This is adapted to gain the best possible solution embedded with the load factor.

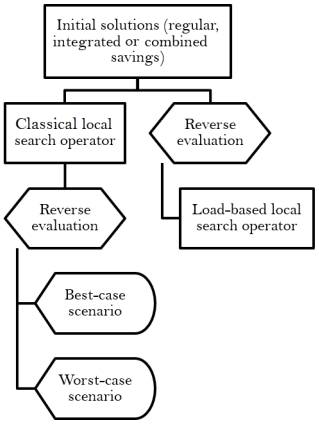

FIGURE 4.1: Way of recording the results with the local search operators

4.2.1 The 2-opt method

For a given R_i, consider customers k and l with $0 < k < l-1$ and $l \leq nc(R_i)$, the route is in the order of $R_i(0, 1, 2, \ldots, k, k+1, \ldots, l, l+1, \ldots, nc(R_i), 0)$. Note that the customer's position of 0 and $nc(R_i)+1$ in the route is the depot. Based on the 2-opt method, the arc of vertex k and $k+1$ is exchanged with the arc of vertex l and $l+1$. Therefore, we can try to reorder the customers to obtain a new tour which is called R_i' with the order of $R_i'(0, 1, 2, \ldots, k, l, \ldots, k+1, l+1, \ldots, nc(R_i), 0)$. It can be observed that if there is a crossing, the procedure will eliminate it and this will lead to a reduced cost of the tour. This procedure is computed formally as follows.

Normally, the distance cost of the route can be obtained by the summation of the distances when travelling from the depot to the customers in the route and back to the depot. The initial total cost is denoted by

$$\begin{aligned}
C_d^0(i) &= d(R_i(0), R_i(1)) + d(R_i(1), R_i(2)) + \cdots + d(R_i(k), R_i(k+1)) \\
&\quad + d(R_i(k+1), R_i(k+2)) + \cdots + d(R_i(l-1), R_i(l)) \\
&\quad + d(R_i(l), R_i(l+1)) + \cdots + d(R_i(nc(R_i)), R_i(0)) \\
&= \sum_{a=0}^{k-1} \sum_{b=a+1}^{k} d(R_i(a), R_i(b)) + d(R_i(k), R_i(k+1)) + \sum_{a=k+1}^{l-1} \sum_{b=a+1}^{l} d(R_i(a), R_i(b)) \\
&\quad + d(R_i(l), R_i(l+1)) + \sum_{a=l+1}^{nc(R_i)} \sum_{b=a+1}^{nc(R_i)+1} d(R_i(a), R_i(b)) \\
&= \sum_{a=0}^{nc(R_i)} \sum_{b=a+1}^{nc(R_i)+1} d(R_i(a), R_i(b)).
\end{aligned}$$

Now, when we reorder the customers in the route according to the 2-opt move, the total cost of the route has to be computed again and is obtained as follows:

$$\begin{aligned}
C_d^1(i) &= d(R_i(0), R_i(1)) + d(R_i(1), R_i(2)) + \cdots + d(R_i(k), R_i(l)) \\
&\quad + d(R_i(l), R_i(l-1)) + \cdots + d(R_i(k+2), R_i(k+1)) \\
&\quad + d(R_i(k+1), R_i(l+1)) + \cdots + d(R_i(nc(R_i)), R_i(0)) \\
&= \sum_{a=0}^{k-1} \sum_{b=a+1}^{k} d(R_i(a), R_i(b)) + d(R_i(k), R_i(l)) + \sum_{a=l}^{k+2} \sum_{b=a-1}^{k+1} d(R_i(a), R_i(b)) \\
&\quad + d(R_i(k+1), R_i(l+1)) + \sum_{a=l+1}^{nc(R_i)} \sum_{b=a+1}^{nc(R_i)+1} d(R_i(a), R_i(b))
\end{aligned}$$

$$= \sum_{a=0}^{nc(R_i)} \sum_{b=a+1}^{nc(R_i)+1} d(R'_i(a), R'_i(b)),$$

where R'_i is introduced as the new route i after the change.

Therefore, we can found the change in cost which is the difference between the old and the new route cost. After basic mathematical manipulation this change in cost $\Delta^d_{kl}(i)$ is reduced to

$$\begin{aligned}\Delta^d_{kl}(i) &= C^0_d(i) - C^1_d(i) \\ &= d(R_i(k), R_i(k+1)) + d(R_i(l), R_i(l+1)) - d(R_i(k), R_i(l)) \\ &\quad - d(R_i(k+1), R_i(l+1)).\end{aligned} \tag{4.1}$$

We refer to $\Delta^d_{kl}(i)$ as the change in cost for the classical 2-opt operator. Note that, if $\Delta^d_{kl}(i) > 0$, there is a gain in implementing such a change. When applying this move, for the customers which are in between vertices $k+1$ and l, the cost or distance between them remains the same although their order in the route is not.

Effect of the load factor

The change in cost based on the load factor is also calculated in the same manner as the distance cost, where the cost of the new route is subtracted from the old route cost but here the cost is expressed also in terms of fuel consumption. The initial fuel consumption cost for route i is given by

$$\begin{aligned}C^0_f(i) &= E(R_i(0), R_i(1)) + E(R_i(1), R_i(2)) + \cdots + E(R_i(k), R_i(k+1)) \\ &\quad + E(R_i(k+1), R_i(k+2)) + \cdots + E(R_i(l-1), R_i(l)) \\ &\quad + E(R_i(l), R_i(l+1)) + \cdots + E(R_i(nc(R_i)), R_i(0)) \\ &= \sum_{a=0}^{k-1} \sum_{b=a+1}^{k} E(R_i(a), R_i(b)) + E(R_i(k), R_i(k+1)) \\ &\quad + \sum_{a=k+1}^{l-1} \sum_{b=a+1}^{l} E(R_i(a), R_i(b)) + E(R_i(l), R_i(l+1)) \\ &\quad + \sum_{a=l+1}^{nc(R_i)} \sum_{b=a+1}^{nc(R_i)+1} E(R_i(a), R_i(b)) \\ &= \sum_{a=0}^{nc(R_i)} \sum_{b=a+1}^{nc(R_i)+1} E(R_i(a), R_i(b)).\end{aligned}$$

The fuel consumption cost for the new reordered route is denoted as

$$\begin{aligned} C_f^1(i) &= E(R_i(0), R_i(1)) + E(R_i(1), R_i(2)) + \cdots + E(R_i(k), R_i(l)) \\ &\quad + E(R_i(l), R_i(l-1)) + \cdots + E(R_i(k+2), R_i(k+1)) \\ &\quad + E(R_i(k+1), R_i(l+1)) + \cdots + E(R_i(nc(R_i)), R_i(0)) \\ &= \sum_{a=0}^{nc(R_i)} \sum_{b=a+1}^{nc(R_i)+1} E(R_i'(a), R_i'(b)). \end{aligned}$$

From there, the outcome of the subtraction will give us the change in cost for the load-based 2-opt operator, $\Delta_{kl}^f(i)$ as follows:

$$\begin{aligned} \Delta_{kl}^f(i) &= C_f^0 - C_f^1 \\ &= \alpha_0[d(R_i(k), R_i(k+1)) + d(R_i(l), R_i(l+1)) - d(R_i(k), R_i(l)) \\ &\quad - d(R_i(k+1), R_i(l+1))] \\ &\quad + \alpha_1[L(R_i(k), R_i(k+1)) \cdot (d(R_i(k), R_i(k+1)) - d(R_i(k), R_i(l))) \\ &\quad + \sum_{a=k+1}^{l-1} \sum_{b=a+1}^{l} (\sum_{c=b}^{l} q(R_i(c)) - \sum_{d=k+1}^{a} q(R_i(d))) \cdot d(R_i(a), R_i(b)) \\ &\quad + L(R_i(l), R_i(l+1)) \cdot (d(R_i(l), R_i(l+1)) - d(R_i(k+1), R_i(l+1)))]. \end{aligned}$$

Substituting the appropriate place with $\Delta_{kl}^d(i)$ as in (4.1) will reduce this formula to

$$\begin{aligned} \Delta_{kl}^f(i) &= \alpha_0[\Delta_{kl}^d(i)] \\ &\quad + \alpha_1[L(R_i(k), R_i(k+1)) \cdot (d(R_i(k), R_i(k+1)) - d(R_i(k), R_i(l))) \\ &\quad + \sum_{a=k+1}^{l-1} \sum_{b=a+1}^{l} (\sum_{c=b}^{l} q(R_i(c)) - \sum_{d=k+1}^{a} q(R_i(d))) \cdot d(R_i(a), R_i(b)) \\ &\quad + L(R_i(l), R_i(l+1)) \cdot (d(R_i(l), R_i(l+1)) - d(R_i(k+1), R_i(l+1)))]. \end{aligned} \tag{4.2}$$

Note that if $\alpha_0 = 1$ and $\alpha_1 = 0$, (4.2) becomes exactly (4.1).

Illustrative example

Consider a network of a tour (0, 1, 2, 3, 4, 5, 6, 7, 8, 0) of eight customers as in Figure 4.2 where the total length and the total amount of fuel consumed are 64.48 and 11.12 units, respectively. With the use of the classical 2-opt operator, the tour is reordered to obtain a new tour (0, 1, 7, 6, 5, 4, 3, 2, 8, 0) with the

total length reduced to 61.28 unit. The percentage deviation of the new and old route cost in terms of its length is approximately 5% with the gain of 3.2 units. The amount of fuel consumed for this new tour is 11.08 unit.

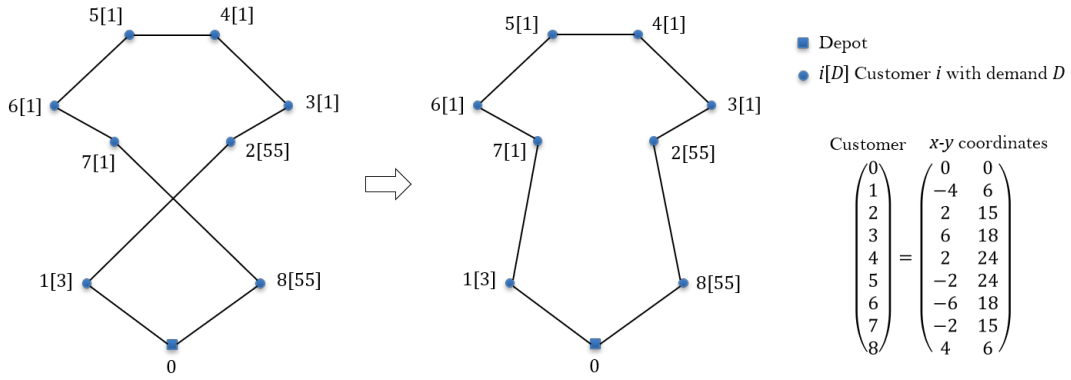

FIGURE 4.2: Sample network of 2-opt move

Whereas, the load-based 2-opt move will reorder the customers on the route according to the fuel consumption cost. As a result, this procedure returns the tour as (0, 8, 2, 3, 4, 5, 6, 7, 1, 0). This tour gives the same length as the classical 2-opt result but in the opposite course with 9.86 units of fuel consumption. Here, although the length of the tour is unchanged, the fuel consumption for the tour is found to be 11% less than the resulted tour of the classical 2-opt move.

Brute-force and formula-based computation

To show the importance of coming up with the new formulation containing the load factor, we conducted a comparative empirical analysis between the brute-force and the formula-based computations in terms of their computational time. The brute force computation is executed based on the concept of finding the difference between the initial and the new route cost after performing such an operator. The computation is performed until the procedure could not improve the solution. Note that there is a large number of redundant computations due to the calculation of certain parts of the route that are unchanged. The formula-based computation is referred to as the newly developed formulation based on the move, which can be either the classical or the load based ones.

Consider this analogy when performing the 2-opt move on route R_1. The number of customers in R_1 is denoted as $nc(R_1)$. If δ is the computing cost of removing two edges and reconnecting the remaining two segments using a full calculation,

then the total computing effort is obtained by

$$\delta \cdot nc(R_1)$$

Now, if δ' is the computing cost of removing two edges and reconnecting the remaining two segments using the formulae (either (4.2) or (4.1)) and $\delta' \ll \delta$, then

$$\delta' \cdot nc(R_1) \ll \delta \cdot nc(R_1)$$

We test the computations on a sample of five instances in each of the small, medium, and large data sets. We record the computing time beginning from when the initial solution from the savings method is acquired until the final solution is formed. Both computations are based on the load-based move which consider the fuel consumption (3.4) of the route to make the move. The formula-based computation is referred to as the new formulation of the load-based 2-opt (4.2). An illustration of the analysis with regard to its average computational time is given in Figure 4.3 where a massive gap of CPU time between the brute-force and the formula-based searches can be observed.

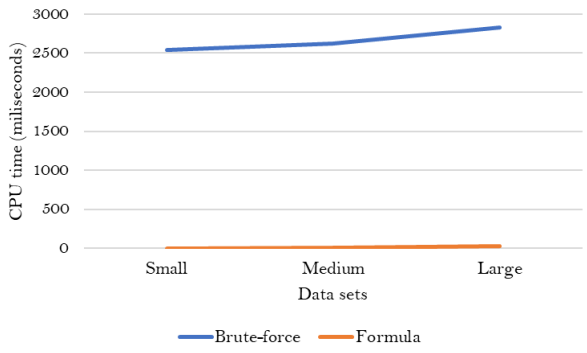

FIGURE 4.3: Average computational time of the load-based 2-opt using brute-force and formula-based computation

Although the brute-force search is simple and uncomplicated to execute and will constantly acquire a solution if it exists, its cost very much depends on the size of the problems. Therefore, the brute-force search is more useful when the problem size is very limited and the ease of implementation is more crucial than the speed. In this study, we show the comparison in terms of computing time to see if it is worth doing the formulae, in case some question its use and may believe it is easier to calculate the full cost before and after the procedure. This fact is very

misleading as the use of the formula is more than 2500 time faster (see Figure 4.3).

Analysis and computational results

In this section, we perform the calculation of percent deviation as in (3.8) where Z is the total objective function obtained when the classical operator is applied to regular savings solutions. While Z_k is obtained when the load-based operator is applied to the regular and the combined savings solutions in which the subscript $k = lr$ for the regular savings and $k = lc$ for the combined savings. The performance of the load-based operators is compared to the best (i.e. LBR and LBC) and the worst-case (i.e. LWR and LWC) scenarios of the solutions following the evaluation schemes given in chapter 3. These two scenarios of solutions are acquired from using the classical operators with the initial solution from the regular savings. This can reflect whether the new load-based operators perform better than the best-case of non-load-based operators or they are still acceptable if they perform better than the worst-case. The analysis is tested on the well-known 46 benchmark VRP test instances given in Table 3.2.

The analysis for the 2-opt operator can be found in Table 4.1 where the average percentage deviation is calculated across five values of $\alpha = (0.6, 0.7, 0.8, 0.9, 1.0)$ for each small, medium and large data sets. When applying the load-based 2-opt to the initial solution obtained from the regular savings, in general, the solutions are improved when compared to both cases of the classical 2-opt. This improvement although is a minor with the highest one at 1.36%, is considered as a success indicating that the incorporation of the load factor within the operator is needed. In contrast, the lowest percent deviation is at -0.29% when $\alpha = 0.9$ for small data set of LBC.

It can therefore be shown that the solutions using the load-based 2-opt from the combined savings are also improved. These are denoted by the positive percent deviation even though we cannot claim the same for the small data set. Note that, when $\alpha = 1$ the same average percent deviation (for LBR and LWR; LBC and LWC) can be seen as this may be because the consideration of the load factor does not exist and the search is based on the distance only.

The analysis is also demonstrated in Figure 4.4. It can be seen in (a) the deviation percentage for medium and large data sets seems to have levelled off but a downward trend can be seen for small data set as well as in (c). Although in (b)

TABLE 4.1: Average percent deviation (in %) of the load-based 2-opt against the classical 2-opt

α	Data set	LBR	LWR	LBC	LWC
0.6	Small	0.24	**1.36**	-0.25	0.21
	Medium	0.31	0.82	0.16	0.74
	Large	0.25	0.93	0.28	1.08
0.7	Small	0.23	1.05	-0.24	0.10
	Medium	0.30	0.68	0.16	0.56
	Large	0.25	0.75	0.48	1.07
0.8	Small	0.22	0.75	-0.23	-0.06
	Medium	0.30	0.55	0.26	0.54
	Large	0.25	0.57	0.31	0.64
0.9	Small	0.20	0.46	**-0.29**	-0.23
	Medium	0.30	0.42	0.26	0.38
	Large	0.25	0.40	0.47	0.67
1.0	Small	0.19	0.19	-0.21	-0.21
	Medium	0.30	0.30	0.26	0.26
	Large	0.24	0.24	0.24	0.24

LBR Load-based operator against the best-case of classical operator using regular savings
LWR Load-based operator against the worst-case of classical operator using regular savings
LBC Load-based operator from combined savings against the best-case of classical operator from regular savings
LWC Load-based operator from combined savings against the worst-case of classical operator from regular savings

and (d), there is no clear trend or similarities between the data sets, it is shown that there are negative percentages in small data set and the fluctuations of the average percent deviation across five values of α are shown for medium and large data sets.

We also present a summary table for the LBC as in Table 4.2 to assess the performance of incorporating the load factor from the initial solution. It can be observed that the average percentage deviation for small data set is similar for all five values of α, although the solutions are all worse than the classical one. This is in contrast with the solutions from medium and large data sets as they all have positive improvement with the highest average percentage of 0.48%. About 75% (or 9 out of 12) of the instances in large data set, which corresponds to the highest average percent deviation, record the highest count of the positive percentage. With regard to the solution that produced negative percent deviation, the small data set

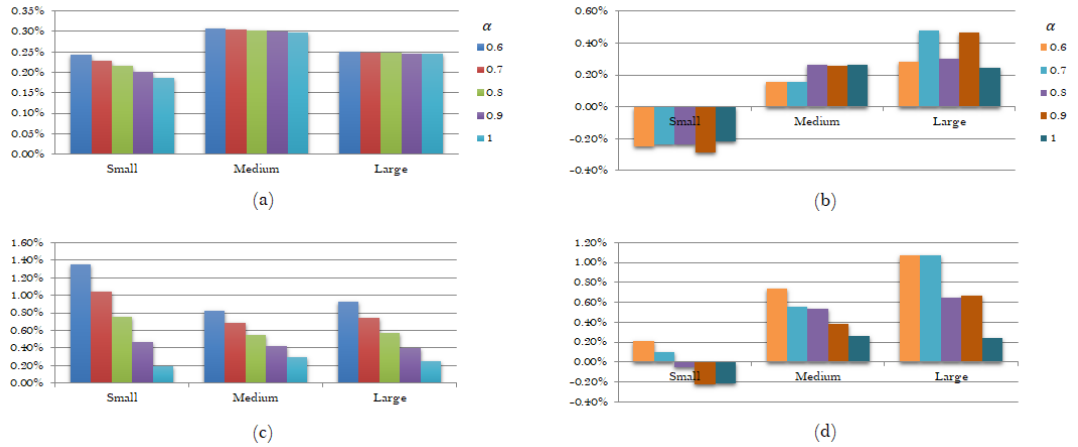

FIGURE 4.4: The average percent deviation of load-based 2-opt against the classical 2-opt; (a) LBR, (b) LBC, (c) LWR, and (d) LWC

records the highest frequency (out of 14, 7 or 50% of the instances). Moreover, the best and the worst percent deviation are 2.13% and -1.88%, respectively.

CPU times

During the improvement phase with the use of the operator, the computing time is logged before commencing the procedure once the initial solution from the savings method is obtained. The time is recorded until the final solution is found. For simplicity, the operator OI(p) represents the classical (i.e. O = C) and the load-based (i.e. O = L) operator p from the initial solution using the regular (i.e. I = R) and the combined (i.e. I = C) savings. Table 4.3 shows the average computational time for 2-opt. Overall, we can say that the load-based 2-opt took almost five times longer than the classical 2-opt. This extra computational burden is expected due to the additional calculations in (4.2).

4.2.2 The (1,0) operator

The (1,0) operator is carried out by taking out a customer from a route and putting it back whether in the same or another route. Inserting back a removed customer in the same route but in a different position is what we meant by intra-route case. We refer to the case where a removed customer is inserted in another route as the inter-route case. Although these two cases are normally combined together, we

TABLE 4.2: Summary table for LBC case of 2-opt

Data set		α				
		0.6	0.7	0.8	0.9	1.0
Small	AvgDev(%)	-0.25	-0.24	-0.23	-0.29	-0.21
(14 instances)	#best	2	3	3	2	1
	#worst	**7**	5	6	**7**	**7**
	Best(%)	0.21	0.14	0.09	0.48	0.34
	Worst(%)	-1.82	-1.84	-1.86	**-1.88**	-1.73
Medium	AvgDev(%)	0.16	0.16	0.26	0.26	0.26
(20 instances)	#best	10	12	11	10	10
	#worst	4	4	3	6	4
	Best(%)	1.18	0.83	1.27	1.08	1.07
	Worst(%)	-0.11	-0.21	-0.11	-0.16	-0.40
Large	AvgDev(%)	0.28	**0.48**	0.31	0.47	0.24
(12 instances)	#best	8	**9**	5	8	8
	#worst	2	2	3	1	2
	Best(%)	1.44	1.93	1.70	**2.13**	1.02
	Worst(%)	-0.31	-0.05	-0.05	-0.02	-0.05

TABLE 4.3: Average computational time for 2-opt (seconds)

Data set	CR(2-opt)	LR(2-opt)	LC(2-opt)
Small	0.000	0.000	0.001
Medium	0.002	0.003	0.005
Large	0.012	0.072	0.080
Overall	0.004	0.020	0.024

CR(p) Classical operator p from regular savings
LR(p) Load-based operator p from regular savings
LC(p) Load-based operator p from combined savings

chose to do it separately to monitor the effect of each move within the route and among the routes.

The first part of the procedure is to compute the cost of removing customer m from route i. This is performed by deducting the new cost of the tour after removing customer m from the old tour cost. Given the old tour as $R_i(0, 1, 2, \ldots, m-1, m, m+1, \ldots, nc(R_i), 0)$ where customer m is within the route. The old cost of the tour in terms of the distance is obtained by adding up the distance travelled along the route from the depot to each customer and back to the depot. Note

again that $nc(R_i)+1$ is the depot 0. This can be shown as

$$\begin{aligned} C_d^0(i) &= d(R_i(0),R_i(1)) + d(R_i(1),R_i(2)) + \cdots + d(R_i(m-1),R_i(m)) \\ &\quad + d(R_i(m),R_i(m+1)) + \cdots + d(R_i(nc(R_i)),R_i(0)) \\ &= \sum_{a=0}^{m-2}\sum_{b=a+1}^{m-1} d(R_i(a),R_i(b)) + d(R_i(m-1),R_i(m)) \\ &\quad + d(R_i(m),R_i(m+1)) + \sum_{a=m+1}^{nc(R_i)}\sum_{b=a+1}^{nc(R_i)+1} d(R_i(a),R_i(b)) \\ &= \sum_{a=0}^{nc(R_i)}\sum_{b=a+1}^{nc(R_i)+1} d(R_i(a),R_i(b)). \end{aligned} \tag{4.3}$$

After removing customer m from route i, a new tour i is created. This is called R_i' where the route is given as $R_i'(0,1,2,\ldots,m-1,m+1,\ldots,nc(R_i'),0)$. The number of customers for R_i' $(nc(R_i'))$ is one less than the number of customers in R_i. The distance cost of this tour is presented as

$$\begin{aligned} C_d^1(i) &= d(R_i(0),R_i(1)) + d(R_i(1),R_i(2)) + \cdots + d(R_i(m-1),R_i(m+1)) \\ &\quad + \cdots + d(R_i(nc(R_i)),R_i(0)) \\ &= d(R_i'(0),R_i'(1)) + d(R_i'(1),R_i'(2)) + \cdots + d(R_i'(nc(R_i)-1),R_i'(0)) \\ &= \sum_{a=0}^{nc(R_i)-1}\sum_{b=a+1}^{nc(R_i)} d(R_i'(a),R_i'(b)). \end{aligned} \tag{4.4}$$

Hence, we subtract the new cost from the old cost to obtain the removal cost of this move. This can be simplified as follows:

$$\begin{aligned} \delta_m^{rd}(i) &= C_d^0(i) - C_d^1(i) \\ &= d(R_i(m-1),R_i(m)) + d(R_i(m),R_i(m+1)) \\ &\quad - d(R_i(m-1),R_i(m+1)). \end{aligned} \tag{4.5}$$

The second part of the procedure is to acquire the cost of inserting back the removed customer m into a different position in the same route i or in another route j. The intra-route and inter-route cases for this operator will be explained later. This procedure is then completed with the subtraction of the insertion cost from the removal cost of the route.

(i) The intra-route of (1,0)

One way to deal with this method is by renaming the revised route R_i' after removal as a temporary route R_i^T ($R_i' = R_i^T$). Then, in R_i^T, the removed customer m is inserted after the k^{th} customer. A new route R_i'' is introduced upon the insertion. Subsequently, the number of customers for the route R_i'' becomes $nc(R_i)$ and the depot 0 is placed at $nc(R_i)+1$. The cost of the route after inserting customer m next to customer k in R_i^T is denoted as

$$\begin{aligned} C_d^2(i) &= d(R_i^T(0), R_i^T(1)) + d(R_i^T(1), R_i^T(2)) + \cdots + d(R_i^T(k), R_i(m)) \\ &\quad + d(R_i(m), R_i^T(k+1)) + \cdots + d(R_i^T(nc(R_i)-1), R_i^T(0)) \\ &= d(R_i''(0), R_i''(1)) + d(R_i''(1), R_i''(2)) + \cdots + d(R_i''(nc(R_i)), R_i''(0)) \\ &= \sum_{a=0}^{nc(R_i)} \sum_{b=a+1}^{nc(R_i)+1} d(R_i''(a), R_i''(b)). \end{aligned}$$

The insertion cost is calculated by subtracting the route cost after removing customer m from the new route cost of inserting it after the k^{th} customer. This is computed as

$$\begin{aligned} \delta_{km}^{id}(i) &= C_d^2(i) - C_d^1(i) \\ &= d(R_i^T(k), R_i(m)) + d(R_i(m), R_i^T(k+1)) \\ &\quad - d(R_i^T(k), R_i^T(k+1)) \end{aligned} \tag{4.6}$$

Thus, we can obtain the change in cost in terms of the distance for the procedure within the route by removing the insertion cost (see (4.6)) from the removal cost (see (4.5)).

$$\begin{aligned} \Delta_{km}^{d}(i) &= \delta_{m}^{rd}(i) - \delta_{km}^{id}(i) \\ &= d(R_i(m-1), R_i(m)) + d(R_i(m), R_i(m+1)) \\ &\quad - d(R_i(m-1), R_i(m+1)) - d(R_i^T(k), R_i(m)) \\ &\quad - d(R_i(m), R_i^T(k+1)) + d(R_i^T(k), R_i^T(k+1)) \end{aligned} \tag{4.7}$$

We refer $\Delta_{km}^{d}(i)$ as the change in cost for the classical intra-route (1,0) operator. A positive value of $\Delta_{km}^{d}(i)$ indicates that a gain of the route cost is achieved when executing such a change.

Effect of the load factor

When adapting the load factor in this operator with regard to the fuel consumption $E(m,n)$ as the route cost, the removal process of customer m from route i is given as the subtraction of the new fuel consumption cost from the old cost of the route. The initial fuel consumption cost of route i is given as follows:

$$\begin{aligned} C_f^0(i) &= E(R_i(0),R_i(1)) + E(R_i(1),R_i(2)) + \cdots + E(R_i(m-1),R_i(m)) \\ &\quad + E(R_i(m),R_i(m+1)) + \cdots + E(R_i(nc(R_i)),R_i(0)) \\ &= \sum_{a=0}^{nc(R_i)} \sum_{b=a+1}^{nc(R_i)+1} E(R_i(a),R_i(b)). \end{aligned} \tag{4.8}$$

Next, the fuel consumption cost for route i is recalculated after the removal of customer m which is presented below.

$$\begin{aligned} C_f^1(i) &= E(R_i(0),R_i(1)) + E(R_i(1),R_i(2)) + \cdots + E(R_i(m-1),R_i(m+1)) \\ &\quad + \cdots + E(R_i(nc(R_i)-1),R_i(0)) \\ &= E(R_i'(0),R_i'(1)) + E(R_i'(1),R_i'(2)) + \cdots + E(R_i'(nc(R_i)-1),R_i'(0)) \\ &= \sum_{a=0}^{nc(R_i)-1} \sum_{b=a+1}^{nc(R_i)} E(R_i'(a),R_i'(b)) \end{aligned} \tag{4.9}$$

Therefore, the removal cost in terms of fuel consumption can be found with the outcome of the deduction.

$$\begin{aligned} \delta_m^{rf}(i) &= C_f^0(i) - C_f^1(i) \\ &= \alpha_0[d(R_i(m-1),R_i(m)) + d(R_i(m),R_i(m+1)) \\ &\quad - d(R_i(m-1),R_i(m+1))] \\ &\quad + \alpha_1[q(R_i(m)) \sum_{a=0}^{m-2} \sum_{b=a+1}^{m-1} d(R_i(a),R_i(b)) \\ &\quad + L(R_i(m-1),R_i(m)) \cdot d(R_i(m-1),R_i(m)) \\ &\quad + L(R_i(m),R_i(m+1)) \cdot (d(R_i(m),R_i(m+1)) \\ &\quad - d(R_i(m-1),R_i(m+1)))] \end{aligned} \tag{4.10}$$

To insert the removed customer m back in the same route i, a temporary route is created where $(R_i' = R_i^T)$. The inserting point is next to the k^{th} customer in

the new temporary route. After the insertion, a new route is introduced and it is denoted as R_i''. Note that, the number of customers for the route R_i'' becomes $nc(R_i)$ and the depot 0 is then located at $nc(R_i)+1$. The following set of equations shows the fuel consumption cost for the new route after the insertion.

$$\begin{aligned} C_f^2(i) &= E(R_i^T(0), R_i^T(1)) + E(R_i^T(1), R_i^T(2)) + \cdots + E(R_i^T(k), R_i(m)) \\ &\quad + E(R_i(m), R_i^T(k+1)) + \cdots + E(R_i^T(nc(R_i)-1), R_i^T(0)) \\ &= E(R_i''(0), R_i''(1)) + E(R_i''(1), R_i''(2)) + \cdots + E(R_i''(nc(R_i)), R_i''(0)) \\ &= \sum_{a=0}^{nc(R_i)} \sum_{b=a+1}^{nc(R_i)+1} E(R_i''(a), R_i''(b)) \end{aligned}$$

The insertion cost for this procedure in terms of fuel consumption can be found when we subtract the $C_f^1(i)$ from the $C_f^2(i)$. This is simplified after mathematical manipulations as follows:

$$\begin{aligned} \delta_{km}^{if}(i) &= C_f^2(i) - C_f^1(i) \\ &= \alpha_0[d(R_i^T(k), R_i(m)) + d(R_i(m), R_i^T(k+1)) \\ &\quad - d(R_i^T(k), R_i^T(k+1))] \\ &\quad + \alpha_1[q(R_i(m)) \sum_{a=0}^{k-1} \sum_{b=a+1}^{k} d(R_i^T(a), R_i^T(b)) \\ &\quad + (L(R_i^T(k), R_i^T(k+1)) + q(R_i(m))) \cdot d(R_i^T(k), R_i(m)) \\ &\quad + L(R_i^T(k), R_i^T(k+1)) \cdot (d(R_i(m), R_i^T(k+1)) \\ &\quad - d(R_i^T(k), R_i^T(k+1)))] \end{aligned} \tag{4.11}$$

Hence, the intra-route procedure of (1,0) operator based on the load factor is completed when the change in cost $\Delta_{km}^f(i)$ is obtained through the subtraction of the insertion cost (4.11) from the removal cost (4.10). We refer $\Delta_{km}^f(i)$ as the change in cost for the load-based intra-route (1,0) operator.

$$\begin{aligned} \Delta_{km}^f(i) &= \delta_m^{rf}(i) - \delta_{km}^{if}(i) \\ &= \alpha_0[d(R_i(m-1), R_i(m)) + d(R_i(m), R_i(m+1)) \\ &\quad - d(R_i(m-1), R_i(m+1)) - d(R_i^T(k), R_i(m)) \\ &\quad - d(R_i(m), R_i^T(k+1)) + d(R_i^T(k), R_i^T(k+1))] \\ &\quad + \alpha_1[q(R_i(m)) \cdot (\sum_{a=0}^{m-2} \sum_{b=a+1}^{m-1} d(R_i(a), R_i(b)) \end{aligned}$$

$$
\begin{aligned}
&- \sum_{c=0}^{k-1} \sum_{d=c+1}^{k} d(R_i^T(c), R_i^T(d))) \qquad (4.12)\\
&+ L(R_i(m-1), R_i(m)) \cdot d(R_i(m-1), R_i(m))\\
&+ L(R_i(m), R_i(m+1)) \cdot (d(R_i(m), R_i(m+1))\\
&\quad - d(R_i(m-1), R_i(m+1)))\\
&- (L(R_i^T(k), R_i^T(k+1)) + q(R_i(m))) \cdot d(R_i^T(k), R_i(m))\\
&- L(R_i^T(k), R_i^T(k+1)) \cdot (d(R_i(m), R_i^T(k+1))\\
&\quad - d(R_i^T(k), R_i^T(k+1)))]
\end{aligned}
$$

Note that when $\alpha_0 = 1$ and $\alpha_1 = 0$, (4.12) reduces to (4.7).

Illustrative example

We consider an example of a route consisting of ten customers as illustrated in Figure 4.5. The initial route $R(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 0)$ has a route length $C_d = 66.22$ and fuel consumption $C_e = 11.53$. The intra-route (1,0) operator shift customer 5 in such a way that the resulted route becomes $R'(0, 1, 2, 3, 4, 6, 7, 8, 9, 5, 10, 0)$. This gives the route length and fuel consumption of $C_d = 63.65$ and $C_e = 11.08$, respectively. Both intra-route operators using the classical and the load-based (1,0) produced similar result. Note that, these operators may begin with different starting solutions (i.e. in the opposite direction) but eventually reach the same final solution. This is because the reverse route evaluation gives the chance for the classical operator to select the route with the direction of the least fuel consumption.

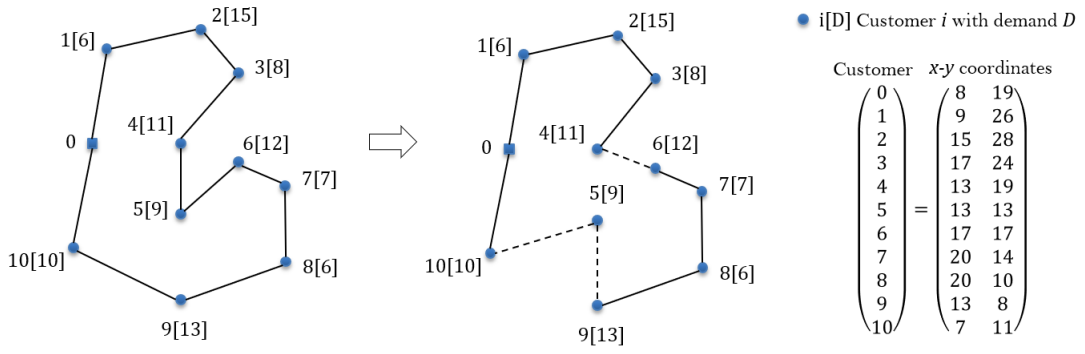

FIGURE 4.5: An example of (1,0) intra-route move

(ii) The inter-route of (1,0)

The case of inter-route (1,0) operator is basically meant to remove a customer from one route and place it into another route. Given that customer m from route i is removed, the removal cost of this move is as shown in (4.5). The next step is to insert $R_i(m)$ in route j next to customer l. Here, we consider route j from the existing route that is established before and also an additional empty route. The intra-route problem can also be combined with the inter-route problem in such a way where the temporary route R_i^T is considered as one of the route j. In this research, we decided to do the intra-route and the inter-route cases separately.

The cost of inserting customer m from route i to route j after the l^{th} customer can be found by subtracting the initial route cost from the new route cost with the insertion. Note that in every tour, the vehicle needs to return to the depot at the end of the tour. The initial cost for the route j is denoted below with the final point of the route is at $nc(R_j)+1$ which is the depot 0.

$$\begin{aligned} C_d^0(j) &= d(R_j(0), R_j(1)) + d(R_j(1), R_j(2)) + \cdots + d(R_j(l), R_j(l+1)) \\ &\quad + \cdots + d(R_j(nc(R_j)), R_j(0)) \\ &= \sum_{a=0}^{nc(R_j)} \sum_{b=a+1}^{nc(R_j)+1} d(R_j(a), R_j(b)) \end{aligned} \tag{4.13}$$

Then, the cost of the new route after the insertion is presented as follows:

$$\begin{aligned} C_d^1(j) &= d(R_j(0), R_j(1)) + d(R_j(1), R_j(2)) + \cdots + d(R_j(l), R_i(m)) \\ &\quad + d(R_i(m), R_j(l+1)) + \cdots + d(R_j(nc(R_j)+1), R_j(0)) \\ &= d(R_j''(0), R_j''(1)) + d(R_j''(1), R_j''(2)) + \cdots + d(R_j''(nc(R_j)+1), R_j''(0)) \\ &= \sum_{a=0}^{nc(R_j)+1} \sum_{b=a+1}^{nc(R_j)+2} d(R_j''(a), R_j''(b)) \end{aligned}$$

where R_j'' is introduced as the new route j once the insertion takes place. The depot is positioned at $nc(R_j)+2$ since the number of customers of the new route R_j'' is added with one to the origin.

The change in cost of R_j when inserting $R_i(m)$ which also refers to the insertion cost is obtained through basic mathematical manipulation.

$$\begin{aligned}\delta^{id}_{lm}(j) &= C^1_d(j) - C^0_d(j) \\ &= d(R_j(l), R_i(m)) + d(R_i(m), R_j(l+1)) - d(R_j(l), R_j(l+1))\end{aligned} \tag{4.14}$$

Subsequently, we subtract the insertion cost (4.14) from the removal cost (4.5) in obtaining the change in cost for this procedure among the routes in terms of the distance travelled. This is performed as follows:

$$\begin{aligned}\Delta^d_{lm}(ij) &= \delta^{rd}_m(i) - \delta^{id}_{lm}(j) \\ &= d(R_i(m-1), R_i(m)) + d(R_i(m), R_i(m+1)) \\ &\quad - d(R_i(m-1), R_i(m+1)) - d(R_j(l), R_i(m)) \\ &\quad - d(R_i(m), R_j(l+1)) + d(R_j(l), R_j(l+1))\end{aligned} \tag{4.15}$$

Effect of the load factor

The inter-route (1,0) operator with the incorporation of the load factor has the same procedure with the classical one but it is following the calculation of vehicle fuel consumption rather than the vehicle distance travelled. The procedure involved the calculation of the removal and the insertion cost. Given that customer m from route i is removed, the removal cost in terms of the fuel consumption can be obtained using (4.10). Afterward, this customer m is inserted in route j next to customer l. The initial fuel consumption cost for route j before the insertion is given by,

$$\begin{aligned}C^0_f(j) &= E(R_j(0), R_j(1)) + E(R_j(1), R_j(2)) + \cdots + E(R_j(l), R_j(l+1)) \\ &\quad + \cdots + E(R_j(nc(R_j)), R_j(0)) \\ &= \sum_{a=0}^{nc(R_j)} \sum_{b=a+1}^{nc(R_j)+1} E(R_j(a), R_j(b))\end{aligned} \tag{4.16}$$

Route j is reordered after the insertion and the fuel consumption cost of the new route j is calculated as

$$\begin{aligned}C^1_f(j) &= E(R_j(0), R_j(1)) + E(R_j(1), R_j(2)) + \cdots + E(R_j(l), R_i(m)) \\ &\quad + E(R_i(m), R_j(l+1)) + \cdots + E(R_j(nc(R_j)+1), R_j(0))\end{aligned}$$

$$= E(R''_j(0), R''_j(1)) + E(R''_j(1), R''_j(2)) + \cdots + E(R''_j(nc(R_j)+1), R''_j(0))$$
$$= \sum_{a=0}^{nc(R_j)+1} \sum_{b=a+1}^{nc(R_j)+2} E(R''_j(a), R''_j(b))$$

The insertion cost with regard to the fuel consumption is obtained by subtracting the initial cost from the new cost of the route and this can be presented as follows:

$$\begin{aligned}
\delta^{if}_{lm}(j) &= C^1_f(j) - C^0_f(j) \\
&= \alpha_0[d(R_j(l), R_i(m)) + d(R_i(m), R_j(l+1)) - d(R_j(l), R_j(l+1))] \\
&\quad + \alpha_1[q(R_i(m)) \sum_{a=0}^{l-1} \sum_{b=a+1}^{l} d(R_j(a), R_j(b)) \\
&\quad + (L(R_j(l), R_j(l+1)) + q(R_i(m))) \cdot d(R_j(l), R_i(m)) \\
&\quad - L(R_j(l), R_j(l+1)) \cdot (d(R_j(l), R_j(l+1)) \\
&\quad - d(R_i(m), R_j(l+1)))]
\end{aligned} \tag{4.17}$$

Finally, we can find the change in fuel consumption cost using the (1,0) operator move among the route by calculating the difference between the removal (4.10) and the insertion (4.17) cost of the route. This is shown along these lines.

$$\begin{aligned}
\Delta^f_{lm}(ij) &= \delta^{rf}_m(i) - \delta^{if}_{lm}(j) \\
&= \alpha_0[d(R_i(m-1), R_i(m)) + d(R_i(m), R_i(m+1)) \\
&\quad - d(R_i(m-1), R_i(m+1)) - d(R_j(l), R_i(m)) \\
&\quad - d(R_i(m), R_j(l+1)) + d(R_j(l), R_j(l+1))] \\
&\quad + \alpha_1[q(R_i(m)) \cdot (\sum_{a=0}^{m-2} \sum_{b=a+1}^{m-1} d(R_i(a), R_i(b)) - \sum_{c=0}^{l-1} \sum_{d=c+1}^{l} d(R_j(c), R_j(d))) \\
&\quad + L(R_i(m-1), R_i(m)) \cdot d(R_i(m-1), R_i(m)) \\
&\quad + L(R_i(m), R_i(m+1)) \cdot (d(R_i(m), R_i(m+1)) \\
&\quad\quad - d(R_i(m-1), R_i(m+1))) \\
&\quad - (L(R_j(l), R_j(l+1)) + q(R_i(m))) \cdot d(R_j(l), R_i(m)) \\
&\quad + L(R_j(l), R_j(l+1)) \cdot (d(R_j(l), R_j(l+1)) - d(R_i(m), R_j(l+1)))]
\end{aligned}$$

Substituting the $\Delta^{d}_{lm}(ij)$ (4.15) in the appropriate place will reduced this formula to,

$$\begin{aligned}
\Delta^{f}_{lm}(ij) &= \delta^{rf}_{m}(i) - \delta^{if}_{lm}(j) \\
&= \alpha_0[\Delta^{d}_{lm}(ij)] \\
&\quad + \alpha_1[q(R_i(m)) \cdot (\sum_{a=0}^{m-2} \sum_{b=a+1}^{m-1} d(R_i(a), R_i(b)) \\
&\qquad - \sum_{c=0}^{l-1} \sum_{d=c+1}^{l} d(R_j(c), R_j(d))) \\
&\quad + L(R_i(m-1), R_i(m)) \cdot d(R_i(m-1), R_i(m)) \\
&\quad + L(R_i(m), R_i(m+1)) \cdot (d(R_i(m), R_i(m+1)) \\
&\qquad - d(R_i(m-1), R_i(m+1))) \\
&\quad - (L(R_j(l), R_j(l+1)) + q(R_i(m))) \cdot d(R_j(l), R_i(m)) \\
&\quad + L(R_j(l), R_j(l+1)) \cdot (d(R_j(l), R_j(l+1)) \\
&\qquad - d(R_i(m), R_j(l+1)))]
\end{aligned} \tag{4.18}$$

Illustrative example

Consider an example consisting of ten customers which are initially assigned to two routes, $R_1(0, 1, 2, 3, 4, 5, 0)$ and $R_2(0, 6, 7, 8, 9, 10, 0)$ as depicted in Figure 4.6. The total length C_d for R_1 and R_2 are 34.34 and 40.93, respectively. The total fuel consumption C_e for R_1 and R_2 are 5.62 and 6.69, respectively. When the operator (1,0) is implemented between the routes, it will try to relocate a customer in another route and find the best gain of making the move. In this example, the best move is to relocate customer 4 from R_1 to R_2, next to customer 6.

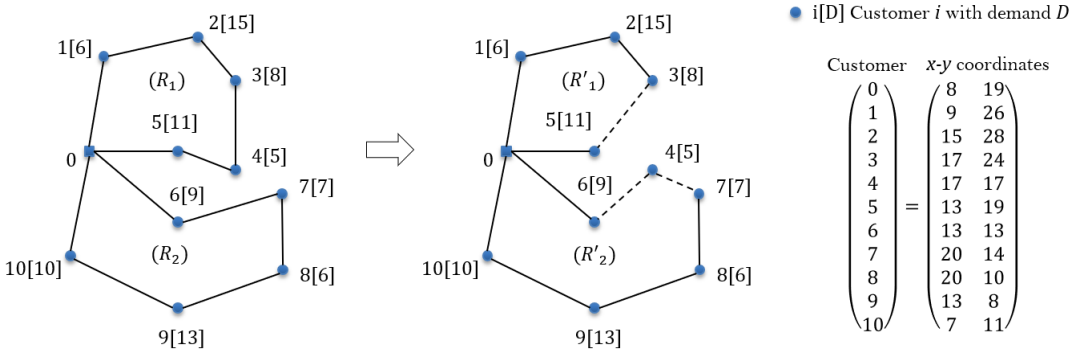

FIGURE 4.6: An example of (1,0) inter-route move

Both the classical and the load-based (1,0) operator decide to make this move which produced two updated routes $R_1'(0, 5, 3, 2, 1, 0)$ and $R_2'(0, 10, 9, 8, 7, 4, 6, 0)$. This new configuration has C_d of 29.27 and 43.76 as well as C_e of 4.71 and 7.16, respectively of R_1' and R_2'. However, the load-based (1,0) commencing the procedure with different direction of the initial solution where $R_1(0, 5, 4, 3, 2, 1, 0)$ and $R_2(0, 10, 9, 8, 7, 6, 0)$. This gives the same total length but different fuel consumption ($C_e(R_1) = 5.57$ and $C_e(R_2) = 6.65$). Given that the reverse route evaluation is applied on a different level (see Figure 4.1) for the classical and the load-based operators, the initial solution before the operator is implemented may be different. The resulted routes may or may not be the same depending on the customer's demand in the particular route. In this example, the resulted routes from using both operators are similar.

Brute-force and formula-based computation

We also highlighted the comparison of brute-force and formula-based computation using the inter-route (1,0) move. Consider an analogy of two routes R_1 and R_2. If δ is the cost of removing a customer from R_1 and replacing it in R_2 using the full calculation then, the computing effort can be acquired by

$$\delta \cdot nc(R_1) \cdot nc(R_2)$$

When using the formulae, the cost of removing a customer from R_1 and replacing it in R_2 is denoted by δ' where $\delta' \ll \delta$. Hence, the computing effort using the formulae can be written as

$$\delta' \cdot nc(R_1) \cdot nc(R_2) \ll \delta \cdot nc(R_1) \cdot nc(R_2)$$

We have performed a computation comparison of the load-based inter-route (1,0) using the brute-force and the formula-based searches. The computation was tested on a sample of five instances in each small, medium and large data sets. The load-based move is based on the fuel consumption function (3.4) and the formula-based search is calculated using (4.18). In the same way as before (load-based 2-opt), the computing time is recorded once the savings method established an initial solution until the operator mechanism developed a final solution. The result of this test is illustrated in Figure 4.7. It can be observed that the difference in terms of computational time is roughly less than 3000 milliseconds (or 3 seconds). We can also observe that the average CPU time for medium data set dropped slightly

from the small data set and then soared up to the large data set.

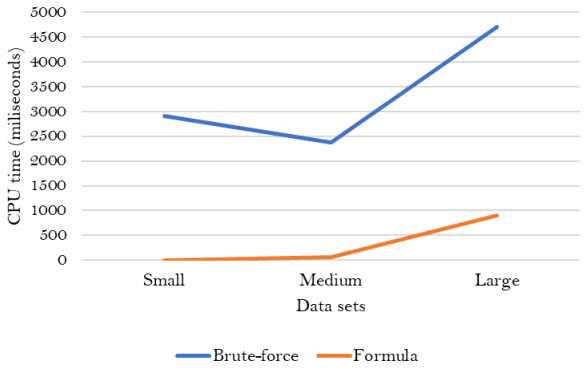

FIGURE 4.7: The average computational time of load-based inter-route (1,0) using the brute-force and the formula-based computations

By using the formula, the search process prevent redundant calculations therefore the time it took to complete the search is significantly reduced. From now on, there is no need to show this comparison for other operators because the massive gap between them is obvious.

Analysis and computational results

The performance of (1,0) relocate operator is demonstrated in Table 4.4 where we show the average percent deviation of the load-based operator against the classical one for intra-route and inter-route cases. Two sources of initial solution namely, the regular and the combined savings are used for this analysis. By using the combined savings as the initial solution along with the load-based operator enable us to observe the effect of load factor from the beginning.

From the table, as a whole we can say that the load-based operator from the combined savings performs better than the one from the regular savings. However, the exception applies for medium data set. This can be seen more clearly in Figures 4.8 and 4.9. Note that, when $\alpha = 1$ the horizontal average percent deviation are all the same for intra-route and inter-route cases. This may be because the load factor is not considered and the search is based on the distance only.

We also show a summary of the result of LBC for intra-route (1,0) operator in Table 4.5. It can be noted that the highest average deviation percentage is at 2.21% for when $\alpha = 0.7$ in the small data set and about 79% (11 out of 14) of the instances produced a positive percentage of improvement. When $\alpha = 0.9$ the best

TABLE 4.4: Average percent deviation (in %) of the load-based (1,0) relocate against the classical (1,0) relocate

α	Data set	Intra-route				Inter-route			
		LBR	LWR	LBC	LWC	LBR	LWR	LBC	LWC
0.6	Small	0.02	1.28	2.06	3.28	-0.05	1.02	1.06	2.12
	Medium	0.10	0.60	-0.49	0.01	0.22	0.73	-0.19	0.32
	Large	0.08	0.77	0.89	1.57	0.09	0.77	1.69	2.36
0.7	Small	0.00	0.92	2.21	3.11	-0.05	0.73	1.81	2.58
	Medium	0.10	0.46	0.00	0.37	0.20	0.57	0.02	0.40
	Large	0.08	0.58	1.23	1.73	0.03	0.54	1.07	1.57
0.8	Small	-0.02	0.58	1.19	1.78	-0.05	0.46	1.46	1.97
	Medium	0.09	0.33	-0.11	0.13	0.19	0.43	-0.09	0.16
	Large	0.08	0.41	0.80	1.12	-0.01	0.31	0.74	1.07
0.9	Small	-0.04	0.26	0.95	1.24	-0.05	0.20	1.35	1.60
	Medium	0.09	0.21	-0.42	-0.30	0.17	0.29	-0.11	0.01
	Large	0.08	0.24	1.04	1.20	-0.06	0.10	0.49	0.65
1.0	Small	-0.06	-0.06	-0.06	-0.06	-0.05	-0.05	-0.05	-0.05
	Medium	0.09	0.09	0.09	0.09	0.16	0.16	0.16	0.16
	Large	0.08	0.08	0.08	0.08	-0.11	-0.11	-0.11	-0.11

percent deviation recorded is at 8.24%. Conversely, the lowest average deviation percentage of -0.49% can be found in medium data set when $\alpha = 0.6$ alongside the worst percent deviation of -6.49%. Out of 20, 13 instances (or 65%) give negative percent deviation when $\alpha = 0.8$ and this represents the highest count for the worst improvement of the operator.

Similarly, a summary result of LBC for inter-route (1,0) relocate can be referred to Table 4.6. In the small data set, the average percent deviation seems to be at the highest (1.81%) when $\alpha = 0.7$. This may correspond to the utmost percent deviation of 7.16%. On the contrary, the load-based inter-route (1,0) from the combined savings recorded the worst percent deviation of -5.36% in the medium data set when compared to the classical operator from the regular savings. In the large data set, the highest frequency (10 out of 12, or about 83% of the instances) of positive and negative percent deviation could be found when $\alpha = 0.6$ and 1.0, respectively.

CPU times

We display the average computational time for small, medium and large data set as well as the overall average when performing the (1,0) relocate operator in Table 4.7. The speed of performing the intra-route (1,0) operator is almost negligible

TABLE 4.5: Summary table for LBC case of intra-route (1,0) relocate

Data set		α				
		0.6	0.7	0.8	0.9	1.0
Small	AvgDev(%)	2.06	**2.21**	1.19	0.95	-0.06
(14 instances)	#best	10	**11**	8	5	1
	#worst	3	2	3	6	3
	Best(%)	8.11	8.16	8.22	**8.24**	0.03
	Worst(%)	-1.31	-1.53	-1.53	-2.87	-0.82
Medium	AvgDev(%)	**-0.49**	0.00	-0.11	-0.42	0.09
(20 instances)	#best	8	9	7	9	5
	#worst	12	11	**13**	11	5
	Best(%)	3.76	3.32	4.09	4.21	1.03
	Worst(%)	**-6.49**	-3.61	-3.61	-4.36	-0.02
Large	AvgDev(%)	0.89	1.23	0.80	1.04	0.08
(12 instances)	#best	8	8	8	8	6
	#worst	4	4	4	4	2
	Best(%)	3.59	3.98	3.27	5.11	0.27
	Worst(%)	-1.69	-1.62	-2.90	-1.68	-0.25

TABLE 4.6: Summary table for LBC case of inter-route (1,0) relocate

Data set		α				
		0.6	0.7	0.8	0.9	1.0
Small	AvgDev(%)	1.06	**1.81**	1.46	1.35	-0.05
(14 instances)	#best	8	10	10	9	2
	#worst	6	4	4	4	5
	Best(%)	5.85	**7.16**	6.02	6.09	0.41
	Worst(%)	-3.00	-2.97	-2.24	-2.32	-0.52
Medium	AvgDev(%)	-0.19	0.02	-0.09	-0.11	0.16
(20 instances)	#best	9	11	11	12	7
	#worst	11	9	9	8	7
	Best(%)	4.33	4.14	2.48	2.17	1.77
	Worst(%)	**-5.36**	-2.84	-2.89	-4.27	-0.29
Large	AvgDev(%)	1.69	1.07	0.74	0.49	-0.11
(12 instances)	#best	**10**	9	7	8	2
	#worst	2	3	5	4	**10**
	Best(%)	4.88	4.10	3.14	3.68	0.18
	Worst(%)	-0.78	-1.31	-3.93	-2.47	-0.51

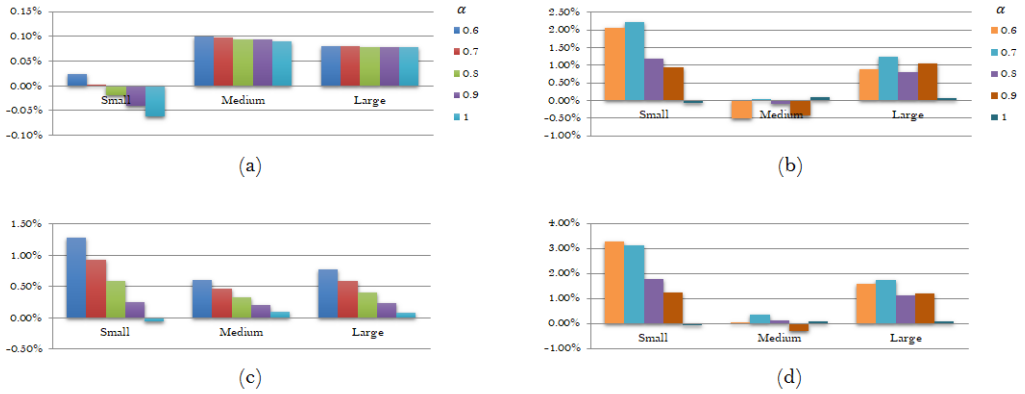

FIGURE 4.8: The average percent deviation of the load-based (1,0) relocate intra-route against the classical (1,0) relocate intra-route; (a) LBR, (b) LBC, (c) LWR, and (d) LWC

where the longest time it took on the overall average is only 8 milliseconds. For the inter-route case, it can be observed that the average CPU time for large data set rises drastically from the medium and small data set.

TABLE 4.7: Average computational time for (1,0) relocate (seconds)

Data set	Intra-route			Inter-route		
	CR(1,0)	LR(1,0)	LC(1,0)	CR(1,0)	LR(1,0)	LC(1,0)
Small	0.000	0.000	0.000	0.000	0.001	0.002
Medium	0.002	0.003	0.004	0.011	0.039	0.045
Large	0.013	0.014	0.024	1.353	2.846	3.056
Overall	0.005	0.005	0.008	0.358	0.759	0.817

4.2.3 The (2,0) operator

This operator is executed by removing two consecutive customers from one route and placing them back in the same route but in different order or in another route. We can divide the procedure of this operator in two phases, the removal and insertion phases. In the first phase, two consecutive customers, namely customers m and $m+1$ are removed from a particular route i. The route cost before and after the removal is recorded and from there we can find the removal cost of this move. The following shows the formula of this move.

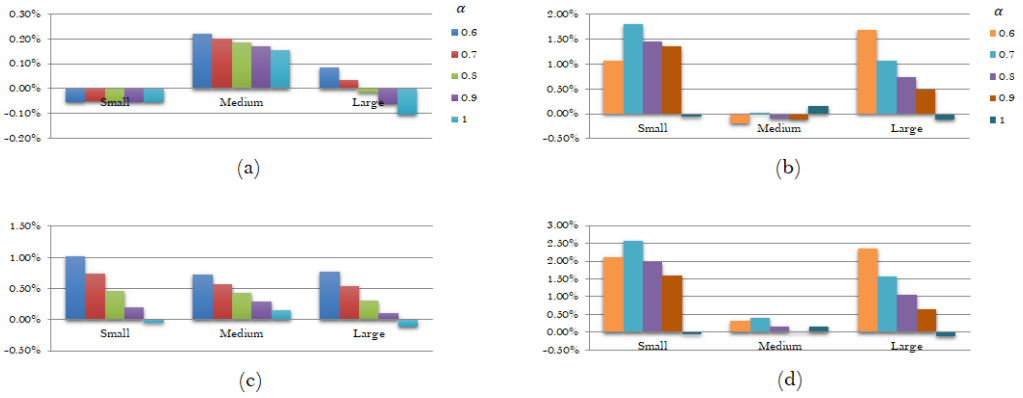

FIGURE 4.9: The average percent deviation of the load-based (1,0) relocate inter-route against the classical (1,0) relocate inter-route; (a) LBR, (b) LBC, (c) LWR, and (d) LWC

We refer to (4.3) as the initial cost of route i. After removing customers m and $m+1$ from route i, customers left in route i are reordered and the new distance cost of the route is expressed. The number of customers in the route is revised to be two less than before.

$$\begin{aligned} C_d^1(i) &= d(R_i(0), R_i(1)) + d(R_i(1), R_i(2)) + \cdots + d(R_i(m-1), R_i(m+2)) \\ &\quad + \cdots + d(R_i(nc(R_i)-2), R_i(0)) \\ &= d(R_i'(0), R_i'(1)) + d(R_i'(1), R_i'(2)) + \cdots + d(R_i'(nc(R_i)-2), R_i'(0)) \\ &= \sum_{a=0}^{nc(R_i)-2} \sum_{b=a+1}^{nc(R_i)-1} d(R_i'(a), R_i'(b)) \end{aligned} \tag{4.19}$$

Hence, we subtract the new cost from the initial cost to obtain the removal cost of this move and this can be simplified as follows:

$$\begin{aligned} \delta_{m^+}^{rd}(i) &= C_d^0(i) - C_d^1(i) \\ &= d(R_i(m-1), R_i(m)) + d(R_i(m), R_i(m+1)) \\ &\quad + d(R_i(m+1), R_i(m+2)) - d(R_i(m-1), R_i(m+2)) \end{aligned} \tag{4.20}$$

The next phase is the process of inserting back the removed customers. Similarly as what we did in the previous operators, the customers are to be inserted in the same route (intra-route) or in another route (inter-route). Although the two cases can be combined together, we chose to perform them separately for better

understanding their individual effect.

(ii) The intra-route (2,0)

A temporary route is created where ($R'_i = R_i^T$) to insert the removed customers m and $m+1$ back in the same route i. The inserting point is next to the k^{th} customer in the new temporary route.

$$\begin{aligned} C_d^2(i) &= d(R_i^T(0), R_i^T(1)) + d(R_i^T(1), R_i^T(2)) + \cdots + d(R_i^T(k), R_i(m)) \\ &\quad + d(R_i(m), R_i(m+1)) + d(R_i(m+1), R_i^T(k+1)) + \ldots \\ &\quad + d(R_i^T(nc(R_i)-2), R_i^T(0)) \\ &= d(R_i''(0), R_i''(1)) + d(R_i''(1), R_i''(2)) + \cdots + d(R_i''(nc(R_i)), R_i''(0)) \\ &= \sum_{a=0}^{nc(R_i)} \sum_{b=a+1}^{nc(R_i)+1} d(R_i''(a), R_i''(b)) \end{aligned}$$

The insertion cost $\delta_{km^+}^{id}(i)$ can be calculated as

$$\begin{aligned} \delta_{km^+}^{id}(i) &= C_d^2(i) - C_d^1(i) \\ &= d(R_i^T(k), R_i(m)) + d(R_i(m), R_i(m+1)) \\ &\quad + d(R_i(m+1), R_i^T(k+1)) - d(R_i^T(k), R_i^T(k+1)) \end{aligned}$$

Thus, the change in cost for the classical intra-route (2,0) operator is formulated as below.

$$\begin{aligned} \Delta_{km^+}^{d}(i) &= \delta_{m^+}^{rd}(i) - \delta_{km^+}^{id}(i) \\ &= d(R_i(m-1), R_i(m)) + d(R_i(m+1), R_i(m+2)) \\ &\quad - d(R_i(m-1), R_i(m+2)) - d(R_i^T(k), R_i(m)) \\ &\quad - d(R_i(m+1), R_i^T(k+1)) + d(R_i^T(k), R_i^T(k+1)) \end{aligned} \tag{4.21}$$

Effect of the load factor

Refer to (4.8) for initial fuel consumption cost for route i. After removing customers m and $m+1$, the fuel consumption cost for route i is revised as written in the following lines.

$$C_f^1(i) = E(R_i(0), R_i(1)) + E(R_i(1), R_i(2)) + \cdots + E(R_i(m-1), R_i(m+2))$$

$$
\begin{aligned}
&+ \cdots + E(R_i(nc(R_i)-2), R_i(0)) \qquad (4.22)\\
&= E(R'_i(0), R'_i(1)) + E(R'_i(1), R'_i(2)) + \cdots + E(R'_i(nc(R_i)-2), R'_i(0))\\
&= \sum_{a=0}^{nc(R_i)-2} \sum_{b=a+1}^{nc(R_i)-1} E(R'_i(a), R'_i(b))
\end{aligned}
$$

The removal cost $\delta^{rf}_{m^+}(i)$ for the load-based intra-route (2,0) relocate can be found as follows.

$$
\begin{aligned}
\delta^{rf}_{m^+}(i) &= C^0_f(i) - C^1_f(i)\\
&= \alpha_0[d(R_i(m-1), R_i(m)) + d(R_i(m), R_i(m+1))\\
&\quad + d(R_i(m+1), R_i(m+2)) - d(R_i(m-1), R_i(m+2))]\\
&\quad + \alpha_1[(q(R_i(m)) + q(R_i(m+1))) \cdot \sum_{a=0}^{m-2} \sum_{b=a+1}^{m-1} d(R_i(a), R_i(b)) \qquad (4.23)\\
&\quad + L(R_i(m-1), R_i(m)) \cdot d(R_i(m-1), R_i(m))\\
&\quad + L(R_i(m), R_i(m+1)) \cdot (d(R_i(m), R_i(m+1))\\
&\quad + L(R_i(m+1), R_i(m+2)) \cdot (d(R_i(m+1), R_i(m+2))\\
&\quad\quad - d(R_i(m-1), R_i(m+2)))]
\end{aligned}
$$

A temporary route is created ($R'_i = R^T_i$) to insert the removed customers m and $m+1$ back in the same route i. The inserting point is next to the k^{th} customer in the new temporary route. The following set of equations shows the fuel consumption cost for the new route after the insertion.

$$
\begin{aligned}
C^2_f(i) &= E(R^T_i(0), R^T_i(1)) + E(R^T_i(1), R^T_i(2)) + \cdots + E(R^T_i(k), R_i(m))\\
&\quad + E(R_i(m), R_i(m+1)) + E(R_i(m+1), R^T_i(k+1)) + \ldots\\
&\quad + E(R^T_i(nc(R_i)-2), R^T_i(0))\\
&= E(R''_i(0), R''_i(1)) + E(R''_i(1), R''_i(2)) + \cdots + E(R''_i(nc(R_i)), R''_i(0))\\
&= \sum_{a=0}^{nc(R_i)} \sum_{b=a+1}^{nc(R_i)+1} E(R''_i(a), R''_i(b))
\end{aligned}
$$

The insertion cost for this procedure, in terms of fuel consumption, can be found when we subtract the $C^1_f(i)$ from the $C^2_f(i)$. After mathematical manipulations,

this is simplified as follows:

$$\begin{aligned}
\delta^{if}_{km^+}(i) &= C^2_f(i) - C^1_f(i) \\
&= \alpha_0[d(R^T_i(k), R_i(m)) + d(R_i(m), R_i(m+1)) \\
&\quad + d(R_i(m+1), R^T_i(k+1)) - d(R^T_i(k), R^T_i(k+1))] \\
&\quad + \alpha_1[(q(R_i(m)) + q(R_i(m+1))) \cdot \sum_{a=0}^{k-1} \sum_{b=a+1}^{k} d(R^T_i(a), R^T_i(b)) \\
&\quad + (L(R^T_i(k), R^T_i(k+1)) + q(R_i(m)) + q(R_i(m+1))) \\
&\quad \cdot d(R^T_i(k), R_i(m)) \\
&\quad + (L(R^T_i(k), R^T_i(k+1)) + q(R_i(m+1))) \cdot d(R_i(m), R_i(m+1)) \\
&\quad + L(R^T_i(k), R^T_i(k+1)) \cdot (d(R_i(m+1), R^T_i(k+1)) \\
&\quad - d(R^T_i(k), R^T_i(k+1)))]
\end{aligned} \tag{4.24}$$

Therefore, the change in cost for the load-based intra-route (2,0) operator which is denoted as $\Delta^f_{km^+}(i)$ can be formulated as follows.

$$\begin{aligned}
\Delta^f_{km^+}(i) &= \delta^{rf}_{m^+}(i) - \delta^{if}_{km^+}(i) \\
&= \alpha_0[d(R_i(m-1), R_i(m)) + d(R_i(m+1), R_i(m+2)) \\
&\quad - d(R_i(m-1), R_i(m+2)) - d(R^T_i(k), R_i(m)) \\
&\quad - d(R_i(m+1), R^T_i(k+1)) + d(R^T_i(k), R^T_i(k+1))] \\
&\quad + \alpha_1[(q(R_i(m)) + q(R_i(m+1))) \cdot (\sum_{a=0}^{m-2} \sum_{b=a+1}^{m-1} d(R_i(a), R_i(b)) \\
&\quad - \sum_{c=0}^{k-1} \sum_{d=c+1}^{k} d(R^T_i(c), R^T_i(d))) \\
&\quad + L(R_i(m-1), R_i(m)) \cdot d(R_i(m-1), R_i(m)) \\
&\quad + L(R_i(m+1), R_i(m+2)) \cdot (d(R_i(m), R_i(m+1)) \\
&\quad + d(R_i(m+1), R_i(m+2)) - d(R_i(m-1), R_i(m+2))) \\
&\quad - (L(R^T_i(k), R^T_i(k+1)) + q(R_i(m)) + q(R_i(m+1))) \\
&\quad \cdot d(R^T_i(k), R_i(m)) \\
&\quad - L(R^T_i(k), R^T_i(k+1)) \cdot (d(R_i(m), R_i(m+1)) \\
&\quad + d(R_i(m+1), R^T_i(k+1)) - d(R^T_i(k), R^T_i(k+1)))]
\end{aligned} \tag{4.25}$$

Illustrative example

To show more clearly how the intra-route (2,0) operator works, we consider an example of eight customers which are assigned to a route $(0, A, B, C, D, E, F, G, H, 0)$ as in Figure 4.10. This route has a total length C_d of 104.87 units and would consume the amount of fuel C_e as 18.95 units. The idea is to move a pair of consecutive customers to another location on the same route. In this example, a chain of customers D and E is relocated next to customer G after performing the operator.

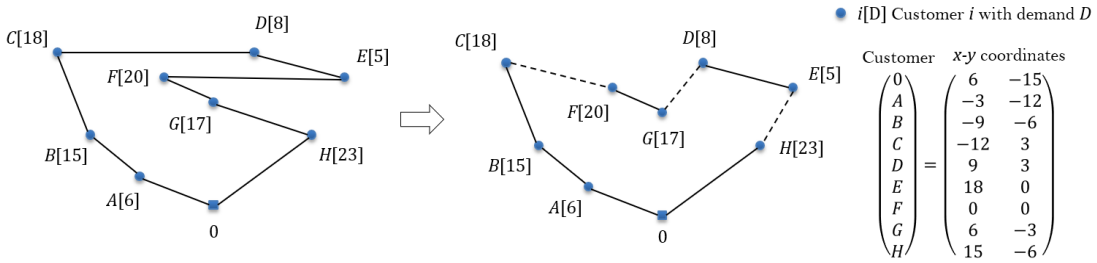

FIGURE 4.10: An example of (2,0) intra-route move

Both the classical and the load-based operators yield a new route $(0, H, E, D, G, F, C, B, A, 0)$ with $C_d = 82.17$ and $C_e = 14.55$. This (2,0) move looks attractive where it returns a gain of 22.7 in length and 4.4 in fuel consumption. Note that, the evaluation of the reverse route gives the other direction of the route as the best one since the original direction has a higher fuel consumption of 14.68 units.

(ii) The inter-route (2,0)

For this operator, we consider the route for insertion including the additional empty route. Refer to (4.13) for the initial cost of route j. The new distance route cost $C_d^1(j)$ is given as

$$\begin{aligned} C_d^1(j) &= d(R_j(0), R_j(1)) + d(R_j(1), R_j(2)) + \cdots + d(R_j(l), R_i(m)) \\ &\quad + d(R_i(m), R_i(m+1)) + d(R_i(m+1), R_j(l+1)) + \ldots \\ &\quad + d(R_j(nc(R_j)+2), R_j(0)) \\ &= d(R'_j(0), R'_j(1)) + d(R'_j(1), R'_j(2)) + \cdots + d(R'_j(nc(R_j)+2), R'_j(0)) \\ &= \sum_{a=0}^{nc(R_j)+2} \sum_{b=a+1}^{nc(R_j)+3} d(R'_j(a), R'_j(b)) \end{aligned} \tag{4.26}$$

The insertion cost can then be found by

$$\begin{aligned}\delta^{id}_{lm^+}(j) &= C^1_d(j) - C^0_d(j) \\ &= d(R_j(l), R_i(m)) + d(R_i(m), R_i(m+1)) \\ &\quad + d(R_i(m+1), R_j(l+1)) - d(R_j(l), R_j(l+1))\end{aligned} \tag{4.27}$$

The change in cost for intra-route (2,0) $\Delta^d_{lm^+}(ij)$ can then be obtained by calculating the difference between the removal (see (4.20)) and the insertion cost.

$$\begin{aligned}\Delta^d_{lm^+}(ij) &= \delta^{rd}_{m^+}(i) - \delta^{id}_{lm^+}(j) \\ &= d(R_i(m-1), R_i(m)) + d(R_i(m+1), R_i(m+2)) \\ &\quad - d(R_i(m-1), R_i(m+2)) - d(R_j(l), R_i(m)) \\ &\quad - d(R_i(m+1), R_j(l+1)) + d(R_j(l), R_j(l+1))\end{aligned} \tag{4.28}$$

Effect of the load factor

The removal process in route i refers to (4.8), (4.22) and (4.23). The initial fuel consumption cost for route j is given as in (4.16). The new route cost in terms of fuel consumption after the insertion of customers m and $m+1$ is given below.

$$\begin{aligned}C^1_f(j) &= E(R_j(0), R_j(1)) + E(R_j(1), R_j(2)) + \cdots + E(R_j(l), R_i(m)) \\ &\quad + E(R_i(m), R_i(m+1)) + E(R_i(m+1), R_j(l+1)) \\ &\quad + \cdots + E(R_j(nc(R_j)+2), R_j(0)) \\ &= E(R''_j(0), R''_j(1)) + E(R''_j(1), R''_j(2)) + \cdots + E(R''_j(nc(R_j)+2), R''_j(0)) \\ &= \sum_{a=0}^{nc(R_j)+2} \sum_{b=a+1}^{nc(R_j)+3} E(R''_j(a), R''_j(b))\end{aligned} \tag{4.29}$$

The insertion cost is derived as follows:

$$\begin{aligned}\delta^{if}_{lm^+}(j) &= C^2_f(j) - C^0_f(j) \\ &= \alpha_0[d(R_j(l), R_i(m)) + d(R_i(m), R_i(m+1)) \\ &\quad + d(R_i(m+1), R_j(l+1)) - d(R_j(l), R_j(l+1))] \\ &\quad + \alpha_1[(q(R_i(m)) + q(R_i(m+1))) \cdot \sum_{a=0}^{l-1} \sum_{b=a+1}^{l} d(R_j(a), R_j(b)) \\ &\quad + (L(R_j(l), R_j(l+1)) + q(R_i(m)) + q(R_i(m+1)))\end{aligned} \tag{4.30}$$

$$
\begin{aligned}
&\cdot d(R_j(l), R_i(m)) \\
&+ (L(R_j(l), R_j(l+1)) + q(R_i(m+1))) \cdot d(R_i(m), R_i(m+1)) \\
&+ L(R_j(l), R_j(l+1)) \cdot (d(R_i(m+1), R_j(l+1)) \\
&\quad - d(R_j(l), R_j(l+1)))]
\end{aligned}
$$

Thus, we can find the change in cost for the load-based inter-route (2,0) operator $\Delta^f_{lm^+}(ij)$ in the following formulation.

$$
\begin{aligned}
\Delta^f_{lm^+}(ij) &= \delta^{rf}_{m^+}(i) - \delta^{if}_{lm^+}(j) \\
&= \alpha_0[d(R_i(m-1), R_i(m)) + d(R_i(m+1), R_i(m+2)) \\
&\quad - d(R_i(m-1), R_i(m+2)) - d(R_j(l), R_i(m)) \\
&\quad - d(R_i(m+1), R_j(l+1)) + d(R_j(l), R_j(l+1))] \\
&\quad + \alpha_1[(q(R_i(m)) + q(R_i(m+1))) \cdot (\sum_{a=0}^{m-2} \sum_{b=a+1}^{m-1} d(R_i(a), R_i(b)) \\
&\quad\quad - \sum_{c=0}^{l-1} \sum_{d=c+1}^{l} d(R_j(c), R_j(d))) \\
&\quad + L(R_i(m-1), R_i(m)) \cdot d(R_i(m-1), R_i(m)) \qquad (4.31) \\
&\quad + L(R_i(m+1), R_i(m+2)) \cdot (d(R_i(m), R_i(m+1)) \\
&\quad\quad + d(R_i(m+1), R_i(m+2)) - d(R_i(m-1), R_i(m+2))) \\
&\quad - (L(R_j(l), R_j(l+1)) + q(R_i(m)) + q(R_i(m+1))) \\
&\quad\quad \cdot d(R_j(l), R_i(m)) \\
&\quad - L(R_j(l), R_j(l+1)) \cdot (d(R_i(m), R_i(m+1)) \\
&\quad\quad + d(R_i(m+1), R_j(l+1)) - d(R_j(l), R_j(l+1)))]
\end{aligned}
$$

Illustrative example

Here we show an inter-route (2,0) move in an example made up of twelve customers which are distributed in two routes $R_1(0, 1, 2, 3, 4, 5, 6, 7, 8, 0)$ and $R_2(0, 9, 10, 11, 12, 0)$. This is illustrated in Figure 4.11. The length of R_1 and R_2 are 29.28 and 19.99 units, respectively while the fuel consumption are 5.11 and 3.26 units respectively for R_1 and R_2. The search will try to move two consecutive customers from one route to another. In this example, a link of customer 1 and 2 from R_1 is chosen to be relocated in R_2 next to customer 9 as this move is considered as the best with the reduction of 2.25 in length and 0.47 in fuel consumption. The new updated routes are $R'_1(0, 3, 4, 5, 6, 7, 8, 0)$ and $R'_2(0, 9, 1, 2, 10, 11, 12, 0)$ which yielded

a length of 22.67 and fuel consumption of 3.81 for R'_1 and a length of 24.35 and fuel consumption of 4.10 for R'_2.

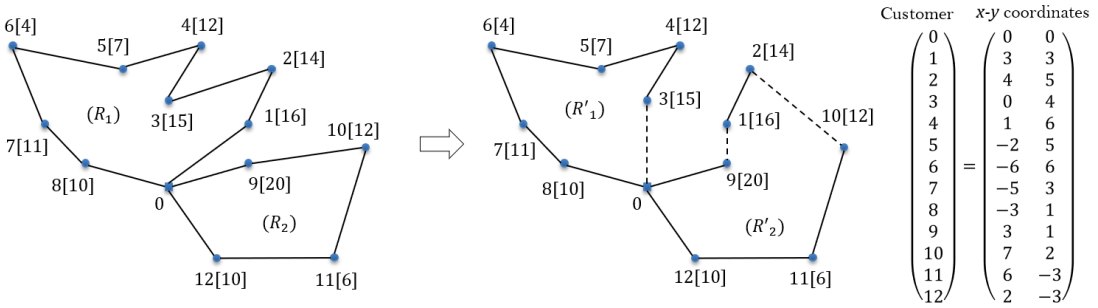

FIGURE 4.11: An example of (2,0) inter-route move

Both the classical and the load-based (2,0) operators gave the same solution configuration. If there are more routes involve in the problem, the configuration may or may not be the same and the fact that we apply the reverse route evaluation on both types of operators increase the chance of the classical operator solution to be as much similar to the load-based ones.

Analysis and computational results

In Table 4.8 we exhibited the average percent deviation of the load-based (2,0) relocate compared with the best and the worst-case of the classical (2,0) relocate (from the regular savings). The load-based (2,0) operator is performed on two initial solutions from the regular and the combined savings. The analysis is implemented on five values of α using the same 46 well-known VRP instances following a similar experiment as with the previous operators. Positive value of deviation percentage means the newly developed load-based operators perform better than the classical ones.

The observation when using this operator is quite similar to the one of (1,0) operator. It can be noted that the percent deviation of the load-based intra-route (2,0) from the combined savings is generally higher than the solution from the regular savings. This is also true for the inter-route case. However, this claim is invalid for the solution from medium data set as the solution using the operator from the regular savings seems to perform better than the combined savings. Additionally, the same percentage can be seen when $\alpha = 1$ for intra-route and inter-route cases. A further illustration of this analysis is shown in Figures 4.12 and 4.13.

TABLE 4.8: Average percent deviation (in %) of the load-based (2,0) relocate against the classical (2,0) relocate

α	Data set	Intra-route				Inter-route			
		LBR	LWR	LBC	LWC	LBR	LWR	LBC	LWC
0.6	Small	-0.02	1.12	2.20	3.31	-0.76	0.51	0.55	1.81
	Medium	0.05	0.56	-0.64	-0.13	-0.05	0.44	-0.53	-0.03
	Large	0.00	0.71	0.82	1.52	0.00	0.70	1.15	1.84
0.7	Small	-0.02	0.81	2.38	3.19	-0.75	0.18	0.69	1.61
	Medium	0.05	0.42	-0.07	0.30	-0.07	0.30	-0.16	0.20
	Large	0.00	0.52	0.99	1.50	-0.01	0.51	1.16	1.67
0.8	Small	-0.03	0.52	1.40	1.94	-0.74	-0.14	0.63	1.22
	Medium	0.05	0.29	-0.18	0.06	-0.08	0.16	-0.24	0.00
	Large	0.00	0.34	0.70	1.03	-0.02	0.32	0.68	1.01
0.9	Small	-0.04	0.23	1.02	1.29	-0.73	-0.44	1.06	1.35
	Medium	0.05	0.17	-0.45	-0.33	-0.09	0.03	-0.52	-0.40
	Large	0.00	0.16	0.70	0.87	-0.02	0.14	0.66	0.83
1.0	Small	-0.04	-0.04	-0.04	-0.04	-0.73	-0.73	-0.73	-0.73
	Medium	0.05	0.05	0.05	0.05	-0.10	-0.10	-0.10	-0.10
	Large	0.00	0.00	0.00	0.00	-0.03	-0.03	-0.03	-0.03

We also provide a summary result of LBC for intra-route (2,0) in Table 4.9. The average percent deviation is found to be the highest when $\alpha = 0.7$ (2.38%) while the lowest at -0.64% when $\alpha = 0.6$. It can also be highlighted the extreme frequencies when there are 11 out of 14 instances or about 79% of the instances in small data set returned a positive percent deviation while 65% of the instances in medium data set returned a negative percentage. The best percent deviation is recorded at 8.70% and the worst one at -6.34%.

Furthermore, a summary table of LBC for inter-route (2,0) can be found in Table 4.10. When $\alpha = 0.6$, it can be seen that the small data set produced the highest count of instances with positive percent deviation (12 out of 14 or about 86% of the instances). With the same value of α, the highest count of instances with negative percent deviation is found in the medium data set (13 out of 20 or 65% of the instances). There are a few observations that can be highlighted when $\alpha = 0.7$. Firstly, the worst percent deviation is recorded at -18.88% in small data set. Secondly, the highest average percent deviation of 1.16% is found in the large data set. In addition, the best percent deviation is obtained in the small data set at 7.08%.

Overall we can say that almost all of the average percent deviation for small and

TABLE 4.9: Summary table for LBC case of intra-route (2,0) relocate

Data set		α				
		0.6	0.7	0.8	0.9	1.0
Small	AvgDev(%)	2.20	**2.38**	1.40	1.02	-0.04
(14 instances)	#best	**11**	**11**	9	7	1
	#worst	3	3	4	4	3
	Best(%)	**8.70**	8.25	8.29	8.38	0.18
	Worst(%)	-2.19	-1.54	-1.54	-3.77	-0.66
Medium	AvgDev(%)	**-0.64**	-0.07	-0.18	-0.45	0.00
(20 instances)	#best	7	10	9	10	5
	#worst	**13**	10	11	10	2
	Best(%)	3.16	2.80	3.44	3.16	0.34
	Worst(%)	**-6.34**	-3.48	-3.51	-4.46	-0.11
Large	AvgDev(%)	0.82	0.99	0.70	0.70	0.00
(12 instances)	#best	8	8	7	9	2
	#worst	4	4	5	3	3
	Best(%)	3.22	3.08	3.24	4.30	0.03
	Worst(%)	-2.52	-1.63	-2.73	-1.69	-0.05

TABLE 4.10: Summary table for LBC case of inter-route (2,0) relocate

Data set		α				
		0.6	0.7	0.8	0.9	1.0
Small	AvgDev(%)	0.55	0.69	0.63	1.06	-0.73
(14 instances)	#best	**12**	11	9	8	2
	#worst	2	3	4	3	3
	Best(%)	4.72	5.19	4.80	**7.08**	0.80
	Worst(%)	-18.72	**-18.88**	-8.42	-4.34	-9.46
Medium	AvgDev(%)	-0.53	-0.16	-0.24	-0.52	-0.10
(20 instances)	#best	7	9	8	10	5
	#worst	**13**	11	12	10	6
	Best(%)	3.34	2.02	2.59	1.63	0.28
	Worst(%)	-6.35	-3.59	-3.29	-4.53	-1.64
Large	AvgDev(%)	1.15	**1.16**	0.68	0.66	-0.03
(12 instances)	#best	9	8	8	8	4
	#worst	3	4	4	4	5
	Best(%)	4.31	3.94	3.19	4.07	0.41
	Worst(%)	-0.81	-0.59	-4.10	-1.61	-0.90

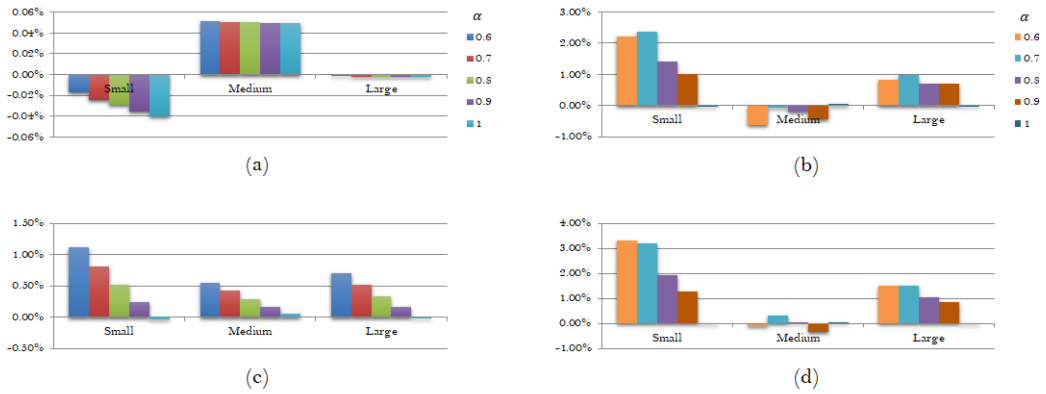

FIGURE 4.12: The average percent deviation of the load-based (2,0) relocate intra-route against the classical (2,0) relocate intra-route; (a) LBR, (b) LBC, (c) LWR, and (d) LWC

large data sets is positive, indicating an improvement of using the load-based (2,0). This is unlikely true when tested on the medium data set.

CPU times

In terms of computing time, we display the result in Table 4.11 where the average values for small, medium and large data sets as well as for the overall data sets are produced. The average computation time in large data set is noticeably higher than the other two sets. This may indicate the search is more complex when performing this operator in the large data set.

TABLE 4.11: Average computational time for (2,0) relocate (seconds)

Data set	Intra-route			Inter-route		
	CR(2,0)	LR(2,0)	LC(2,0)	CR(2,0)	LR(2,0)	LC(2,0)
Small	0.000	0.000	0.000	0.000	0.000	0.001
Medium	0.002	0.003	0.004	0.004	0.008	0.007
Large	0.012	0.052	0.045	0.412	0.565	0.625
Overall	0.004	0.015	0.013	0.109	0.151	0.166

4.3 Summary

In this chapter, we presented three commonly used local search operators, which are 2-opt, (1,0) relocate and (2,0) relocate. The 2-opt operator is performed on

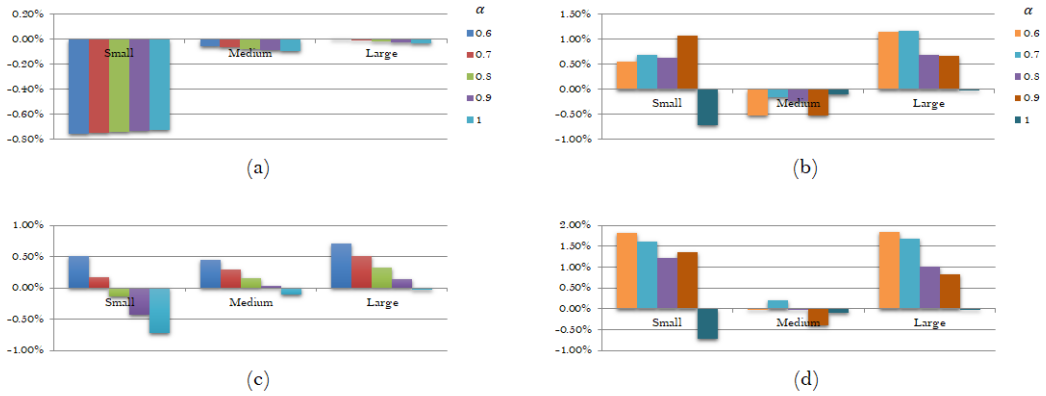

FIGURE 4.13: The average percent deviation of the load-based (2,0) relocate inter-route against the classical (2,0) relocate inter-route; (a) LBR, (b) LBC, (c) LWR, and (d) LWC

intra-route while the other two are performed on both intra-route and inter-route. We introduced the load factor into the formulation and showed it mathematically for each operator. An illustrative example of the move is provided for each operator. The comparative analysis of the load-based operators against the classical operators is tested on the well-known 46 VRP test problems. In addition, we highlighted the differences of the brute-force and the formula-based computation to show the importance of coming up with a new formulae embedded with the load factor. In the next chapter, three other local search operators which we consider to be advanced due to their complexity will be analysed.

Chapter 5

Incorporation of load factor in advanced local search operators

5.1 Introduction

In this chapter, we put forward three more local search operators namely, (1,1), (2,1) and (2,2) interchanges. These advanced operators are considered to be relatively more complicated to compute than those discussed in chapter 4. These lead to spend more time to complete the search. These operators are adapted with the load factor and performed for both intra-route and inter-route cases. We present their formulations mathematically and show an illustrative example for each case. This is followed by an extensive computational test experiment which is undertaken using 46 well-known VRP instances.

5.2 Design of the new operators

One more type of local search operators namely (μ, ψ) interchange is analysed in this chapter. We examine the case of where μ and ψ values are 1 and 2. We consider two customers consecutively when μ or (and) ψ are 2. We have investigated three variants of the (μ, ψ) interchange operator. These include:

(a) Swap or direct exchange: This is the case when m^{th} customer from route i and n^{th} customer from route j are removed, and their exact positions are exchanged. In other words, the position of customer m in route i is replaced

with the position of customer n from route j and the position of customer n in route j is then replaced with the one of customer m from route i.

(b) Best exchange: After customer m and n are removed from route i and j, respectively, customer m is allowed to be placed anywhere in route j and the same for customer n in route i. Here, the customer is placed in the best position so that the maximum cost saving is found. The swap variant that is explained in (a) is the special case where the insertion points refer to $k = m - 1$ and $l = n - 1$, respectively.

(c) Perturb case: This variant is extended when a customer m is removed from route i, and then inserted in route j. Afterward, route i will receive a customer n, which can be from route j or another route, say route k. Therefore, instead of two routes, this procedure could involve three routes when customer n is taken from route k to be inserted in route i. This was initially given by Salhi and Rand (1987) and it is known as perturb.

In this chapter, each operator is implemented and evaluated separately. This is performed within the route and among the routes. For the procedure within the route or intra-route case, we thought that the swap or direct exchange technique given in (a) is the best way to represent the operator as the other variants are similar. For example, the intra-route of (1,1) interchange operator using the best exchange move (b) is similar to the intra-route of (2,0) relocate operator whether it involves two customers in a chain or individually. On the other hand, we follow the best exchange technique as in (b) for the procedure amongst the routes or also known as an inter-route case where the swap method is also included. It may also be interesting to examine the extended version 'perturb' though in this research only the first two variants are implemented.

5.3 The (1,1) interchange operator

The (1,1) operator is basically meant to carry out the process of removing a customer from one route, say route i, which will then receive a customer whether from the same route i or any other route. Now, the removed customer from route i needs to be inserted either back in route i but in the different position or in the best position in any other route.

5.3.1 The intra-route (1,1)

Swap or direct exchange version is considered for the execution of this operator. The classical version in carrying out the (1,1) move within the route is based on the distance measure for the particular route. Consider the position of customer m in route i is exchanged with customer n and the cost of the route before and after the exchange process are expressed as follows:

The initial cost of route i where customers m and n are, is

$$\begin{aligned} C_d^0(i) &= d(R_i(0), R_i(1)) + d(R_i(1), R_i(2)) + \cdots + d(R_i(m-1), R_i(m)) \\ &\quad + d(R_i(m), R_i(m+1)) + \cdots + d(R_i(n-1), R_i(n)) \\ &\quad + d(R_i(n), R_i(n+1)) + \cdots + d(R_i(nc(R_i)), R_i(0)) \\ &= \sum_{a=0}^{nc(R_i)} \sum_{b=a+1}^{nc(R_i)+1} d(R_i(a), R_i(b)) \end{aligned} \tag{5.1}$$

Then, the new route cost after exchanging m and n is rewritten as $C_d^1(i)$.

$$\begin{aligned} C_d^1(i) &= d(R_i(0), R_i(1)) + d(R_i(1), R_i(2)) + \cdots + d(R_i(m-1), R_i(n)) \\ &\quad + d(R_i(n), R_i(m+1)) + \cdots + d(R_i(n-1), R_i(m)) \\ &\quad + d(R_i(m), R_i(n+1)) + \cdots + d(R_i(nc(R_i)), R_i(0)) \\ &= \sum_{a=0}^{nc(R_i)} \sum_{b=a+1}^{nc(R_i)+1} d(R_i'(a), R_i'(b)) \end{aligned}$$

Thus, we can find the difference of the route cost which will produce the change in cost when applying the (1,1) operator within the route is found. Let $\Delta_{mn}^d(i)$ be the change in cost for the classical intra-route (1,1) operator. This is expressed as follows:

$$\begin{aligned} \Delta_{mn}^d(i) &= C_d^0(i) - C_d^1(i) \\ &= d(R_i(m-1), R_i(m)) + d(R_i(m), R_i(m+1)) \\ &\quad + d(R_i(n-1), R_i(n)) + d(R_i(n), R_i(n+1)) \\ &\quad - d(R_i(m-1), R_i(n)) - d(R_i(n), R_i(m+1)) \\ &\quad - d(R_i(n-1), R_i(m)) - d(R_i(m), R_i(n+1)) \end{aligned} \tag{5.2}$$

For the special case if $n = m + 1$ where customer n is situated next to customer m, the new route cost after the exchange will be rewritten in the following.

$$\begin{aligned} C_d^1(i) &= d(R_i(0), R_i(1)) + d(R_i(1), R_i(2)) + \cdots + d(R_i(m-1), R_i(m+1)) \\ &\quad + d(R_i(m+1), R_i(m)) + d(R_i(m), R_i(m+2)) \\ &\quad + \cdots + d(R_i(nc(R_i)), R_i(0)) \\ &= \sum_{a=0}^{nc(R_i)} \sum_{b=a+1}^{nc(R_i)+1} d(R_i'(a), R_i'(b)) \end{aligned}$$

This will then result in a reduced formula of $\Delta_{mn}^d(i)$ as

$$\begin{aligned} \Delta_{mn}^d(i) &= C_d^0(i) - C_d^1(i) \\ &= d(R_i(m-1), R_i(m)) + d(R_i(m+1), R_i(m+2)) \\ &\quad - d(R_i(m-1), R_i(m+1)) - d(R_i(m), R_i(m+2)) \end{aligned}$$

Effect of the load factor

Instead of calculating the route cost in terms of distance travelled, we incorporate the load factor in the routing problem by measuring the amount of fuel consumed by the vehicle when travelling along the particular route. The same idea as the distance cost is used to determine the cost of the route in terms of fuel consumption.

The old fuel consumption cost of route i is given as

$$\begin{aligned} C_f^0(i) &= E(R_i(0), R_i(1)) + E(R_i(1), R_i(2)) + \cdots + E(R_i(m-1), R_i(m)) \\ &\quad + E(R_i(m), R_i(m+1)) + \cdots + E(R_i(n-1), R_i(n)) \\ &\quad + E(R_i(n), R_i(n+1)) + \cdots + E(R_i(nc(R_i)), R_i(0)) \\ &= \sum_{a=0}^{nc(R_i)} \sum_{b=a+1}^{nc(R_i)+1} E(R_i(a), R_i(b)) \end{aligned} \tag{5.3}$$

The exchange between customers m and n will then produce the fuel consumption cost in the following manner.

$$\begin{aligned} C_f^1(i) &= E(R_i(0), R_i(1)) + E(R_i(1), R_i(2)) + \cdots + E(R_i(m-1), R_i(n)) \\ &\quad + E(R_i(n), R_i(m+1)) + \cdots + E(R_i(n-1), R_i(m)) \end{aligned}$$

$$
\begin{aligned}
&+ E(R_i(m), R_i(n+1)) + \cdots + E(R_i(nc(R_i)), R_i(0)) \\
&= \sum_{a=0}^{nc(R_i)} \sum_{b=a+1}^{nc(R_i)+1} E(R'_i(a), R'_i(b))
\end{aligned}
$$

The difference between the previous fuel consumption cost will then give us the change in cost for the load-based intra-route (1,1) operator. This is given below. Note that the expression in the bracket that is multiplied with α_0 is equivalent to $\Delta^d_{mn}(i)$ given in (5.2).

$$
\begin{aligned}
\Delta^f_{mn}(i) &= C^0_f(i) - C^1_f(i) \\
&= \alpha_0[d(R_i(m-1), R_i(m)) + d(R_i(m), R_i(m+1)) \\
&\quad + d(R_i(n-1), R_i(n)) + d(R_i(n), R_i(n+1)) \\
&\quad - d(R_i(m-1), R_i(n)) - d(R_i(n), R_i(m+1)) \\
&\quad - d(R_i(n-1), R_i(m)) - d(R_i(m), R_i(n+1))] \\
&\quad + \alpha_1[L(R_i(m-1), R_i(m)) \cdot (d(R_i(m-1), R_i(m)) \\
&\qquad - d(R_i(m-1), R_i(n))) \\
&\quad + L(R_i(m), R_i(m+1)) \cdot d(R_i(m), R_i(m+1)) \\
&\quad - (L(R_i(m-1), R_i(m)) - q(R_i(n))) \cdot d(R_i(n), R_i(m+1))) \\
&\quad + (q(R_i(n)) - q(R_i(m))) \sum_{a=m+1}^{n-2} \sum_{b=a+1}^{n-1} d(R_i(a), R_i(b)) \\
&\quad + L(R_i(n-1), R_i(n)) \cdot d(R_i(n-1), R_i(n)) \\
&\quad - (L(R_i(n), R_i(n+1)) + q(R_i(m))) \cdot (d(R_i(n-1), R_i(m)) \\
&\quad + L(R_i(n), R_i(n+1)) \cdot (d(R_i(n), R_i(n+1)) \\
&\qquad - d(R_i(m), R_i(n+1)))]
\end{aligned}
$$

For the case if $n = m + 1$, the change in cost for the load-based intra-route (1,1), $\Delta^f_{mn}(i)$ will reduce to the formula as described below.

$$
\begin{aligned}
\Delta^f_{mn}(i) &= \alpha_0[d(R_i(m-1), R_i(m)) + d(R_i(m+1), R_i(m+2)) \\
&\quad - d(R_i(m-1), R_i(m+1)) - d(R_i(m+1), R_i(m+2))] \\
&\quad + \alpha_1[L(R_i(m-1), R_i(m)) \cdot (d(R_i(m-1), R_i(m)) \\
&\qquad - d(R_i(m-1), R_i(m+1))) \\
&\quad + (q(R_i(m+1)) - q(R_i(m))) \cdot d(R_i(m), R_i(m+1)) \\
&\quad + L(R_i(m+1), R_i(m+2)) \cdot (d(R_i(m+1), R_i(m+2))
\end{aligned}
$$

$$- d(R_i(m), R_i(m+2)))]$$

Illustrative example

To show the intra-route move of the (1,1) interchange operator, a small example made up of six customers is displayed in Figure 5.1. Initially, the route is established as $R(0, 1, 2, 3, 4, 5, 6, 0)$ with a length of 81.10 units and fuel consumption of 14.23 units. The procedure will search a profitable move to swap two customers in the route. Following the implementation of the classical as well as the load-based (1,1) intra-route operators, the same final solution $R'(0, 1, 2, 5, 4, 3, 6, 0)$ is formed. This move of swapping the position of customers 3 and 5 results in a reduced total length and fuel consumption of 73.93 and 13.03 units, respectively.

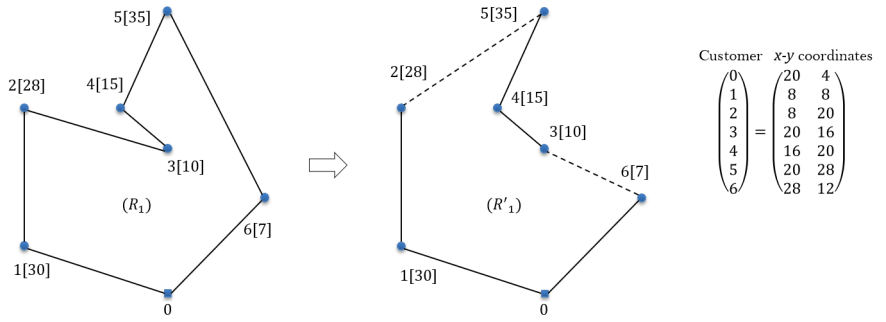

FIGURE 5.1: An example of (1,1) intra-route move

5.3.2 The inter-route (1,1)

The (1,1) operator for the procedure among the routes is executed according to the best exchange variant. The procedure as mentioned before takes place when two customers are removed, one from route i and the other from route j which are then inserted back in route j and route i, respectively. When customer m is removed from route i, the initial and the new cost of the route after the removal of these two customers are shown as in (4.3) and (4.4). A similar calculation for the initial and the new route cost in terms of the distance when customer n is removed from route j are presented below.

$$\begin{aligned} C_d^0(j) = {} & d(R_j(0), R_j(1)) + d(R_j(1), R_j(2)) + \cdots + d(R_j(n-1), R_j(n)) \\ & + d(R_j(n), R_j(n+1)) + \cdots + d(R_j(nc(R_j)), R_j(0)) \end{aligned} \tag{5.4}$$

$$= \sum_{a=0}^{nc(R_j)} \sum_{b=a+1}^{nc(R_j)+1} d(R_j(a), R_j(b))$$

$$\begin{aligned} C_d^1(j) &= d(R_j(0), R_j(1)) + d(R_j(1), R_j(2)) + \cdots + d(R_j(n-1), R_j(n+1)) \\ &\quad + \cdots + d(R_j(nc(R_j)-1), R_j(0)) \\ &= \sum_{a=0}^{nc(R_j)-1} \sum_{b=a+1}^{nc(R_j)} d(R'_j(a), R'_j(b)) \end{aligned} \tag{5.5}$$

The removal cost of the routes is then calculated by deducting the new cost from the initial cost. This is shown in the following lines.

$$\begin{aligned} \delta_{mn}^{rd}(ij) &= C_d^0(i) - C_d^1(i) + C_d^0(j) - C_d^1(j) \\ &= d(R_i(m-1), R_i(m)) + d(R_i(m), R_i(m+1)) \\ &\quad - d(R_i(m-1), R_i(m+1)) + d(R_j(n-1), R_j(n)) \\ &\quad + d(R_j(n), R_j(n+1)) - d(R_j(n-1), R_j(n+1)) \end{aligned} \tag{5.6}$$

The next part of the procedure is the insertion process where $R_i(m)$ is inserted in R'_j after l^{th} customer and $R_j(n)$ is inserted in R'_i after k^{th} customer. The cost of the route for both routes i and j when the insertion is carried out can be found as follows. This is performed in the same way as we calculate the distance cost of the route. This is achieved by accumulating the travelling distance from the depot to the first customer and to the next customer in the established order for the route and lastly back to the depot. For route i, the new cost with the insertion of $R_j(n)$ is as below.

$$\begin{aligned} C_d^2(i) &= d(R'_i(0), R'_i(1)) + d(R'_i(1), R'_i(2)) + \cdots + d(R'_i(k), R_j(n)) \\ &\quad + d(R_j(n), R'_i(k+1)) + \cdots + d(R'_i(nc(R_i)), R'_i(0)) \\ &= \sum_{a=0}^{nc(R_i)} \sum_{b=a+1}^{nc(R_i)+1} d(R''_i(a), R''_i(b)) \end{aligned} \tag{5.7}$$

A similar expression for route j when $R_i(m)$ is inserted next to customer l is presented as follows:

$$\begin{aligned} C_d^2(j) &= d(R'_j(0), R'_j(1)) + d(R'_j(1), R'_j(2)) + \cdots + d(R'_j(l), R_i(m)) \\ &\quad + d(R_i(m), R'_j(l+1)) + \cdots + d(R'_j(nc(R_j)), R'_j(0)) \end{aligned}$$

$$= \sum_{a=0}^{nc(R_j)} \sum_{b=a+1}^{nc(R_j)+1} d(R''_j(a), R''_j(b))$$

The route R'' is introduced as the new route after the insertion.

The insertion cost for this procedure in terms of distance is the difference between C_d^2 and C_d^1 for both routes i and j. This is shown in the following manner.

$$\begin{aligned}\delta^{id}_{knlm}(ij) &= C_d^2(i) - C_d^1(i) + C_d^2(j) - C_d^1(j) \\ &= d(R'_i(k), R_j(n)) + d(R_j(n), R'_i(k+1)) \\ &\quad - d(R'_i(k), R'_i(k+1)) + d(R'_j(l), R_i(m)) \\ &\quad + d(R_i(m), R'_j(l+1)) - d(R'_j(l), R'_j(l+1))\end{aligned} \tag{5.8}$$

Subsequently, we can acquire the change in cost when the inter-route (1,1) operator is applied by subtracting the insertion cost from the removal cost. This is performed along these lines.

$$\begin{aligned}\Delta^{d}_{knlm}(ij) &= \delta^{rd}_{mn}(ij) - \delta^{id}_{knlm}(ij) \\ &= d(R_i(m-1), R_i(m)) + d(R_i(m), R_i(m+1)) \\ &\quad - d(R_i(m-1), R_i(m+1)) + d(R_j(n-1), R_j(n)) \\ &\quad + d(R_j(n), R_j(n+1)) - d(R_j(n-1), R_j(n+1)) \\ &\quad - d(R'_i(k), R_j(n)) - d(R_j(n), R'_i(k+1)) \\ &\quad + d(R'_i(k), R'_i(k+1)) - d(R'_j(l), R_i(m)) \\ &\quad - d(R_i(m), R'_j(l+1)) + d(R'_j(l), R'_j(l+1))\end{aligned} \tag{5.9}$$

Effect of the load factor

We refer to (4.8) and (4.9) of chapter 4 for the initial and the new cost of fuel consumption for route i after removing customer m. The same manner is applied to route j to find the initial (C_f^0) and the new fuel consumption cost (C_f^1) once customer n is removed and this is given below as

$$\begin{aligned}C_f^0(j) &= E(R_j(0), R_j(1)) + E(R_j(1), R_j(2)) + \cdots + E(R_j(n-1), R_j(n)) \\ &\quad + E(R_j(n), R_j(n+1)) + \cdots + E(R_j(nc(R_j)), R_j(0)) \\ &= \sum_{a=0}^{nc(R_j)} \sum_{b=a+1}^{nc(R_j)+1} E(R_j(a), R_j(b))\end{aligned} \tag{5.10}$$

$$\begin{aligned} C_f^1(j) &= E(R_j(0), R_j(1)) + E(R_j(1), R_j(2)) + \dots \\ &\quad + E(R_j(n-1), R_j(n+1)) + \cdots + E(R_j(nc(R_j)-1), R_j(0)) \\ &= \sum_{a=0}^{nc(R_j)-1} \sum_{b=a+1}^{nc(R_j)} E(R'_j(a), R'_j(b)) \end{aligned} \tag{5.11}$$

Next, the reduction of the new cost from the initial cost of both routes i and j will give us the removal cost of fuel consumption for the (1,1) move among the routes. This is denoted as $\delta_{mn}^{rf}(ij)$.

$$\begin{aligned} \delta_{mn}^{rf}(ij) &= C_f^0(i) - C_f^1(i) + C_f^0(j) - C_f^1(j) \\ &= \alpha_0[d(R_i(m-1), R_i(m)) + d(R_i(m), R_i(m+1)) \\ &\quad - d(R_i(m-1), R_i(m+1)) + d(R_j(n-1), R_j(n)) \\ &\quad + d(R_j(n), R_j(n+1)) - d(R_j(n-1), R_j(n+1))] \\ &\quad + \alpha_1[q(R_i(m)) \sum_{a=0}^{m-2} \sum_{b=a+1}^{m-1} d(R_i(a), R_i(b)) \\ &\quad + L(R_i(m-1), R_i(m)) \cdot d(R_i(m-1), R_i(m)) \\ &\quad + L(R_i(m), R_i(m+1)) \cdot (d(R_i(m), R_i(m+1)) \\ &\quad\quad - d(R_i(m-1), R_i(m+1))) \\ &\quad + q(R_j(n)) \sum_{a=0}^{n-2} \sum_{b=a+1}^{n-1} d(R_j(a), R_j(b)) \\ &\quad + L(R_j(n-1), R_j(n)) \cdot d(R_j(n-1), R_j(n)) \\ &\quad + L(R_j(n), R_j(n+1)) \cdot (d(R_j(n), R_j(n+1)) \\ &\quad\quad - d(R_j(n-1), R_j(n+1)))] \end{aligned} \tag{5.12}$$

Another part of the procedure is to exchange the route for customer m and n and place them in such a way that customer m is inserted next to customer l in R'_j and customer n is inserted next to customer k in R'_i. After the insertion, the fuel consumption cost for both routes is revised and they are conveyed as follows. Hence, the new fuel consumption cost for route i with the insertion of $R_j(n)$ is given by

$$\begin{aligned} C_f^2(i) &= E(R'_i(0), R'_i(1)) + E(R'_i(1), R'_i(2)) + \cdots + E(R'_i(k), R_j(n)) \\ &\quad + E(R_j(n), R'_i(k+1)) + \cdots + E(R'_i(nc(R_i)), R'_i(0)) \\ &= \sum_{a=0}^{nc(R_i)} \sum_{b=a+1}^{nc(R_i)+1} E(R''_i(a), R''_i(b)) \end{aligned} \tag{5.13}$$

A similar expression for route j when $R_i(m)$ is inserted next to customer l is shown as

$$\begin{aligned} C_f^2(j) &= E(R'_j(0), R'_j(1)) + E(R'_j(1), R'_j(2)) + \cdots + E(R'_j(l), R_i(m)) \\ &\quad + E(R_i(m), R'_j(l+1)) + \cdots + E(R'_j(nc(R_j)), R'_j(0)) \\ &= \sum_{a=0}^{nc(R_j)} \sum_{b=a+1}^{nc(R_j)+1} E(R''_j(a), R''_j(b)) \end{aligned}$$

We can now find the cost of insertion for the (1,1) move amongst routes i and j by reducing the fuel consumption cost before and after the insertion. It is denoted as $\delta^{if}_{mnkl}(ij)$ and is expressed in the following.

$$\begin{aligned} \delta^{if}_{knlm}(ij) &= C_f^2(i) - C_f^1(i) + C_f^2(j) - C_f^1(j) \\ &= \alpha_0[d(R'_i(k), R_j(n)) + d(R_j(n), R'_i(k+1)) \\ &\quad - d(R'_i(k), R'_i(k+1)) + d(R'_j(l), R_i(m)) \\ &\quad + d(R_i(m), R'_j(l+1)) - d(R'_j(l), R'_j(l+1))] \\ &\quad + \alpha_1[q(R_j(n)) \sum_{a=0}^{k-1} \sum_{b=a+1}^{k} d(R'_i(a), R'_i(b)) \\ &\quad + (L(R'_i(k), R'_i(k+1)) + q(R_j(n))) \cdot d(R'_i(k), R_j(n)) \\ &\quad + L(R'_i(k), R'_i(k+1)) \cdot (d(R_j(n), R'_i(k+1)) \\ &\quad\quad - d(R'_i(k), R'_i(k+1))) \\ &\quad + q(R_i(m)) \sum_{a=0}^{l-1} \sum_{b=a+1}^{l} d(R'_j(a), R'_j(b)) \\ &\quad + (L(R'_j(l), R'_j(l+1)) + q(R_i(m))) \cdot d(R'_j(l), R_i(m)) \\ &\quad + L(R'_j(l), R'_j(l+1)) \cdot (d(R_i(m), R'_j(l+1)) \\ &\quad\quad - d(R'_j(l), R'_j(l+1)))] \end{aligned} \tag{5.14}$$

Upon completion of this procedure, we subtract the insertion cost from the removal cost to obtain the change in cost for the load-based inter-route (1,1) operator, $\Delta^f_{mnkl}(ij)$ which is presented in the subsequent lines.

$$\begin{aligned} \Delta^f_{knlm}(ij) &= \delta^{rf}_{mn}(ij) - \delta^{if}_{knlm}(ij) \\ &= \alpha_0[d(R_i(m-1), R_i(m)) + d(R_i(m), R_i(m+1)) \\ &\quad - d(R_i(m-1), R_i(m+1)) + d(R_j(n-1), R_j(n)) \end{aligned}$$

$$
\begin{aligned}
&+ d(R_j(n), R_j(n+1)) - d(R_j(n-1), R_j(n+1)) \\
&- d(R'_i(k), R_j(n)) - d(R_j(n), R'_i(k+1)) \\
&+ d(R'_i(k), R'_i(k+1)) - d(R'_j(l), R_i(m)) \\
&- d(R_i(m), R'_j(l+1)) + d(R'_j(l), R'_j(l+1))] \\
&+ \alpha_1[q(R_i(m)) \cdot (\sum_{a=0}^{m-2} \sum_{b=a+1}^{m-1} d(R_i(a), R_i(b)) \\
&\quad - \sum_{a=0}^{l-1} \sum_{b=a+1}^{l} d(R'_j(a), R'_j(b))) \\
&+ L(R_i(m-1), R_i(m)) \cdot d(R_i(m-1), R_i(m)) \\
&+ L(R_i(m), R_i(m+1)) \cdot (d(R_i(m), R_i(m+1)) \\
&\quad - d(R_i(m-1), R_i(m+1))) \\
&+ q(R_j(n)) \cdot (\sum_{a=0}^{n-2} \sum_{b=a+1}^{n-1} d(R_j(a), R_j(b)) \\
&\quad - \sum_{a=0}^{k-1} \sum_{b=a+1}^{k} d(R'_i(a), R'_i(b))) \\
&+ L(R_j(n-1), R_j(n)) \cdot d(R_j(n-1), R_j(n)) \\
&+ L(R_j(n), R_j(n+1)) \cdot (d(R_j(n), R_j(n+1)) \\
&\quad - d(R_j(n-1), R_j(n+1))) \\
&- (L(R'_i(k), R'_i(k+1)) + q(R_j(n))) \cdot d(R'_i(k), R_j(n)) \\
&- L(R'_i(k), R'_i(k+1)) \cdot (d(R_j(n), R'_i(k+1)) \\
&\quad - d(R'_i(k), R'_i(k+1))) \\
&- (L(R'_j(l), R'_j(l+1)) + q(R_i(m))) \cdot d(R'_j(l), R_i(m)) \\
&- L(R'_j(l), R'_j(l+1)) \cdot (d(R_i(m), R'_j(l+1)) \\
&\quad - d(R'_j(l), R'_j(l+1)))]
\end{aligned}
\tag{5.15}
$$

Illustrative example

Consider an example of ten customers which are assigned to two routes $R_1(0, 1, 2, 3, 4, 5, 0)$ and $R_2(0, 6, 7, 8, 9, 10, 0)$. This is illustrated in Figure 5.2. These two routes R_1 and R_2 have a length of 75.52 and 84.17 units and consume 12.64 and 13.88 units of fuel respectively. The search of exchanging a customer between these two routes resulted in exchanging customer 4 from R_1 and customer 7 from R_2. This move leads to two new updated routes which are $R'_1(0, 5, 7, 3, 2, 1, 0)$ and $R'_2(0, 6, 4, 8, 9, 10, 0)$. These two routes have a total length of 79.92 and 77.36

units and consume 13.11 and 12.65 units of fuel, respectively. Additionally, this move yields a reduction of 2.41 units in length and 0.76 units of fuel.

Similar to the classical (1,1) inter-route operator, the load-based operator produced the same configuration of the resulted routes. However, it begin the search with the opposite direction for route 1 where $R_1(0, 5, 4, 3, 2, 1, 0)$ and the fuel consumption of 12.27 units.

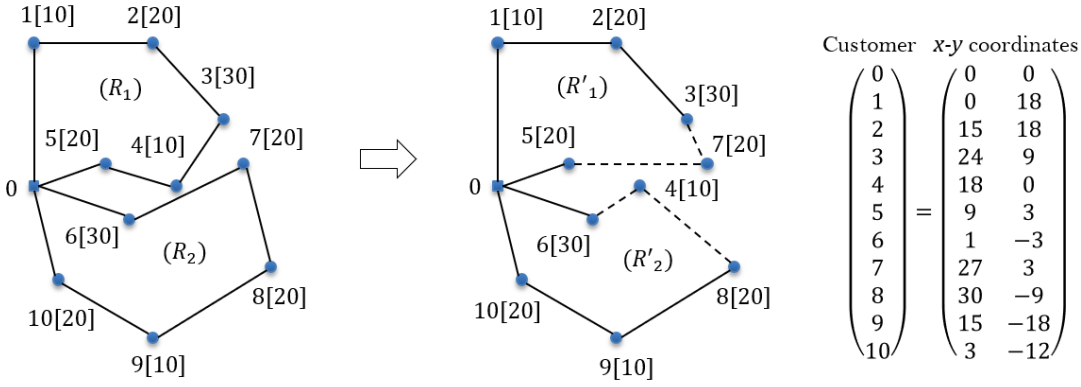

FIGURE 5.2: An example of (1,1) inter-route move

5.3.3 Analysis and computational results

A similar analysis to the one conducted in the previous chapter is also performed here for these operators. The performance of the load-based operators is measured through the comparison of the performance of the classical operators. This is achieved by the calculation of the percentage deviation as in (3.8). The criterion is used to see how much the solution of the load-based operators differs from a known solution of the classical operators using an initial solution from the regular savings. We compare the load-based operator solution to two cases of the classical operator solution. These include the best and the worst-case following the reverse route evaluation as highlighted in section 4.2. The load-based operator is performed on two sets of the initial solution. One is from the regular savings and the other from the combined savings. The analysis is summarised using the following measures.

LBR: The load-based operator compared to the *best*-case of the classical operator. Both use the initial solution from the *regular* savings

LBC: The load-based operator from the *combined* savings compared to the *best*-case of the classical operator from the regular savings

LWR: The load-based operator compared to the *worst*-case of the classical operator. Both use the initial solution from the *regular* savings

LWC: The load-based operator from the *combined* savings compared to the *worst*-case of the classical operator from the regular savings.

Table 5.1 demonstrates the average percent deviation for the load-based (1,1) interchange against the best and the worst case of the classical (1,1) interchange operator. The data showed the result for both intra-route and inter-route cases using $\alpha = (0.6, 0.7, 0.8, 0.9, 1.0)$ in small, medium and large data sets.

It can be observed in the first column (i.e. LBR for intra-route), the average percent deviation using small, medium and large data sets remained constant for different values of α. In addition, the average percent deviation in small data set for all α values is 0%. This may indicate that while using the regular savings to obtain the initial solutions, the load-based and the best-case of classical (1,1) intra-route operators produced an identical solution. It can also be seen a negative percentage for some values of α in the medium data set for intra-route case when the combined savings is used as their initial solution. An illustration of the analysis for the intra-route case is shown in Figure 5.3.

TABLE 5.1: Average percent deviation (in %) of the load-based against the classical (1,1) interchange operator

α	Data set	Intra-route				Inter-route			
		LBR	LWR	LBC	LWC	LBR	LWR	LBC	LWC
0.6	Small	0.00	1.12	2.16	3.25	-0.30	-0.34	-0.04	1.04
	Medium	0.27	0.78	-0.36	0.16	-0.16	0.41	-0.33	0.25
	Large	0.10	0.79	0.81	1.49	-0.16	0.48	1.40	2.02
0.7	Small	0.00	0.82	2.35	3.15	-0.30	-0.34	1.14	1.92
	Medium	0.27	0.64	0.11	0.49	-0.17	0.25	0.06	0.48
	Large	0.10	0.60	0.97	1.47	-0.20	0.27	0.68	1.14
0.8	Small	0.00	0.54	1.28	1.81	-0.30	-0.34	0.51	1.02
	Medium	0.27	0.51	-0.05	0.20	-0.18	0.09	-0.16	0.11
	Large	0.10	0.43	0.68	1.01	-0.24	0.06	0.44	0.73
0.9	Small	0.00	0.26	1.02	1.28	-0.31	-0.33	0.35	0.59
	Medium	0.27	0.39	-0.24	-0.12	-0.19	-0.05	-0.54	-0.41
	Large	0.10	0.26	0.78	0.94	-0.28	-0.13	0.13	0.28
1.0	Small	0.00	0.00	0.00	0.00	-0.31	-0.33	-0.33	-0.33
	Medium	0.27	0.27	0.27	0.27	-0.20	-0.20	-0.20	-0.20
	Large	0.10	0.10	0.10	0.10	-0.31	-0.31	-0.31	-0.31

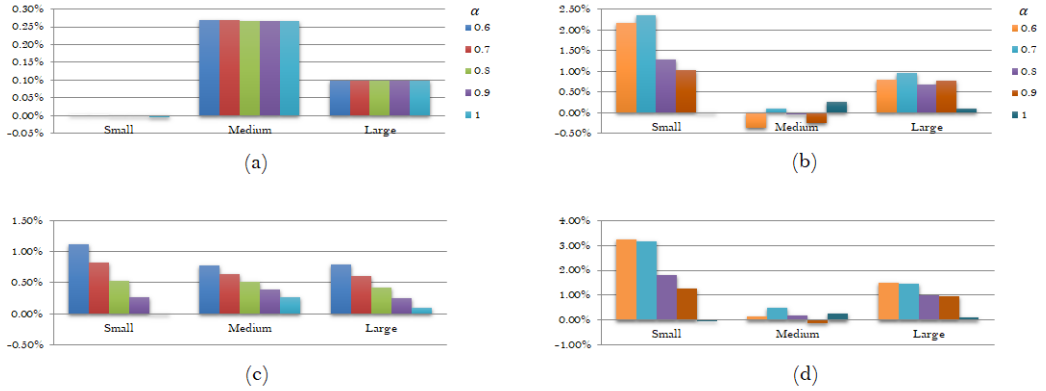

FIGURE 5.3: Average percent deviation of the load-based against the classical (1,1) interchange intra-route operator; (a) LBR, (b) LBC, (c) LWR, and (d) LWC

In terms of the inter-route case, the load-based (1,1) operator from the regular savings seems to perform worst than the best-case of the classical one. This is indicated by the negative percentage in LBR column for all values of α. Looking at the LBC column, most negative percentages resulted from the medium data set even though we can observe some improvement in small and large data sets. Further illustration of the inter-route analysis is shown in Figure 5.4.

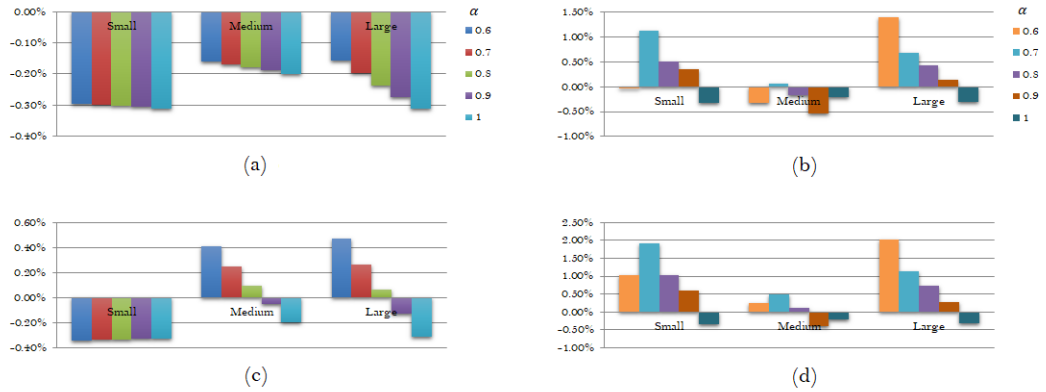

FIGURE 5.4: Average percent deviation of the load-based against the classical (1,1) interchange inter-route operator; (a) LBR, (b) LBC, (c) LWR, and (d) LWC

To observe more thoroughly the performance of (1,1) operator by incorporating the load factor from the beginning, we summarize the result of LBC for both the intra-route and the inter-route cases which can be found in Tables 5.2 and 5.3. There are five aspects in the small, medium and large data sets for each value of α. First, the average percent deviation (AvgDev) followed by the count of positives

(#best) and negatives (#worst) percent deviation. Next, the highest and the lowest percent deviation which is denoted as Best(%) and Worst(%).

With regard to the intra-route, the highest average percent deviation can be found in the small data set when $\alpha = 0.7$. This corresponds to the highest count of positive percent deviation in a data set where about 79% of the instances (or 11 out of 14) in the small data set record a positive percent deviation. While at $\alpha = 0.9$ we can see the best percent deviation of 8.30% in the same data set. On the contrary, the worst deviation percentage can be found in the medium data set at -6.28% when $\alpha = 0.6$ alongside the lowest average percent deviation which is recorded at -0.36%. Out of 20, 11 of the instances (or 55%) have a negative percentage and this is the largest #worst in a data set.

TABLE 5.2: Summary table for LBC case of intra-route (1,1)

Data set		α				
		0.6	0.7	0.8	0.9	1.0
Small	AvgDev(%)	2.16	**2.35**	1.28	1.02	0.00
(14 instances)	#best	**11**	**11**	9	7	0
	#worst	3	3	3	5	1
	Best(%)	7.96	8.09	8.19	**8.30**	0.00
	Worst(%)	-2.00	-1.50	-1.49	-3.22	-0.05
Medium	AvgDev(%)	**-0.36**	0.11	-0.05	-0.24	0.27
(20 instances)	#best	9	10	9	10	11
	#worst	**11**	10	**11**	10	0
	Best(%)	3.58	2.39	2.57	3.75	0.90
	Worst(%)	**-6.28**	-3.41	-3.41	-4.25	0.00
Large	AvgDev(%)	0.81	0.97	0.68	0.78	0.10
(12 instances)	#best	8	7	8	9	6
	#worst	4	5	4	2	2
	Best(%)	3.90	2.84	2.78	4.03	0.25
	Worst(%)	-2.15	-1.50	-2.83	-1.57	-0.12

We refer to the summary table for LBC of inter-route (1,1) interchange in Table 5.3. In the small data set, the best and the worst deviation percentages of 9.34% and -13.46%, respectively are obtained when $\alpha = 0.9$ and 0.6. When $\alpha = 0.8$, 10 of 14 or about 71% of the instances have a negative percent deviation and this is the highest frequency recorded. In the medium data set, we can observe the lowest average percent deviation at -0.54% when $\alpha = 0.9$. Finally in the large data set, it can be noted that the highest average deviation percentage of 1.40% corresponds to the highest count of positive percent deviation where 11 out of 12 (or about 92%) instances have a positive percentage.

TABLE 5.3: Summary table for LBC case of inter-route (1,1)

Data set		α				
		0.6	0.7	0.8	0.9	1.0
Small	AvgDev(%)	-0.04	1.14	0.51	0.35	-0.33
(14 instances)	#best	5	5	4	6	1
	#worst	9	9	**10**	7	8
	Best(%)	9.01	9.12	9.22	**9.34**	0.19
	Worst(%)	**-13.46**	-2.38	-2.34	-2.30	-3.26
Medium	AvgDev(%)	-0.33	0.06	-0.16	**-0.54**	-0.20
(20 instances)	#best	10	9	9	9	7
	#worst	10	11	11	11	12
	Best(%)	2.15	2.01	2.13	1.42	0.49
	Worst(%)	-4.85	-3.76	-3.84	-5.06	-3.09
Large	AvgDev(%)	**1.40**	0.68	0.44	0.13	-0.31
(12 instances)	#best	**11**	9	8	5	3
	#worst	1	3	4	7	9
	Best(%)	2.85	2.58	2.46	4.01	0.75
	Worst(%)	-1.74	-2.42	-3.50	-2.33	-1.38

CPU times

The computational time recorded when using the local search operators in this chapter is similar to the one in the previous chapter. We record the time once the initial solution from the savings method is formed before starting the refinement procedure using the operator. The end time is logged when the final solution is established. The average computational time for the small, medium and large data sets are presented in Table 5.4. The average computational time inclusive for all 46 instances is also given in the table. For simplicity, the operator is rewritten as OI(p) which represents the classical (O=C) and the load-based (O=L) operator p from the initial solution using the regular (I=R) and the combined (I=C) savings.

TABLE 5.4: Average computational time for (1,1) (seconds)

Data set	Intra-route			Inter-route		
	CR(1,1)	LR(1,1)	LC(1,1)	CR(1,1)	LR(1,1)	LC(1,1)
Small	0.000	0.000	0.000	0.050	0.298	0.559
Medium	0.002	0.003	0.004	13.654	152.152	145.853
Large	0.013	0.057	0.082	2308.223	2358.126	2636.024
Overall	0.004	0.016	0.023	608.097	681.407	751.243

It can be shown the overall average computational time for intra-route case records

4, 16 and 23 milliseconds respectively for CR(1,1), LR(1,1) and LC(1,1). For inter-route case, the overall average computing time for CR(1,1), LR(1,1) and LC(1,1) is logged at 608, 681 and 751 seconds, respectively. It can be noted that the average CPU times for inter-route procedure in the large data set seems to shoot up drastically. This may indicate a more difficult search for the large data set compared to the other two. This also show that the new formula are more complex than their counterparts that do not include the load factor.

5.4 The (2,1) interchange operator

For this operator, two consecutive customers from one route is exchanged with a customer from another route. The insertion position after the exchange could be the same as before or in a new location which will yield the least cost. In this study, we perform the move based on the swap variant for the intra-route and the best exchange variant for the inter-route case. The formulations involved in the procedure is described below.

5.4.1 The intra-route (2,1)

The initial route cost for where customers m, $m+1$ and n are in route i can be referred to (5.1). Then, the position of customers m and $m+1$ is exchanged with the position of customer n. The cost of the route is revised and this is presented as follows.

$$\begin{aligned} C_d^1(i) = {} & d(R_i(0), R_i(1)) + d(R_i(1), R_i(2)) + \cdots + d(R_i(m-1), R_i(n)) \\ & + d(R_i(n), R_i(m+2)) + \cdots + d(R_i(n-1), R_i(m)) \\ & + d(R_i(m), R_i(m+1)) + d(R_i(m), R_i(n+1)) \\ & + \cdots + d(R_i(nc(R_i)), R_i(0)) \\ = {} & \sum_{a=0}^{nc(R_i)} \sum_{b=a+1}^{nc(R_i)+1} d(R_i'(a), R_i'(b)) \end{aligned}$$

The change in cost for the classical intra-route (2,1) operator, which is denoted as $\Delta^{d}_{m^{+}n}(i)$, is expressed as follows:

$$\begin{aligned}\Delta^{d}_{m^{+}n}(i) &= C^{0}_{d}(i) - C^{1}_{d}(i) \\ &= d(R_i(m-1), R_i(m)) + d(R_i(m+1), R_i(m+2)) \\ &\quad + d(R_i(n-1), R_i(n)) + d(R_i(n), R_i(n+1)) \\ &\quad - d(R_i(m-1), R_i(n)) - d(R_i(n), R_i(m+2)) \\ &\quad - d(R_i(n-1), R_i(m)) - d(R_i(m+1), R_i(n+1))\end{aligned} \tag{5.16}$$

There are two conditions related to where m and n are situated which could reduce the change in cost $\Delta^{d}_{m^{+}n}(i)$.

(a) Case 1: $n > m$, if $n = m + 2$ then

$$\begin{aligned}\Delta^{d}_{m^{+}n}(i) &= d(R_i(m-1), R_i(m)) + d(R_i(m+1), R_i(m+2)) \\ &\quad + d(R_i(m+2), R_i(m+3)) - d(R_i(m-1), R_i(m+2)) \\ &\quad - d(R_i(m+2), R_i(m)) - d(R_i(m+1), R_i(m+3))\end{aligned}$$

(b) Case 2: $n < m$, if $n = m - 1$ then

$$\begin{aligned}\Delta^{d}_{m^{+}n}(i) &= d(R_i(m-2), R_i(m-1)) + d(R_i(m-1), R_i(m)) \\ &\quad + d(R_i(m+1), R_i(m+2)) - d(R_i(m-2), R_i(m)) \\ &\quad - d(R_i(m+1), R_i(m-1)) - d(R_i(m-1), R_i(m+2))\end{aligned}$$

Effect of the load factor

The route cost is expressed in terms of fuel consumption rather than in distance only. The initial fuel consumption cost for route i can be referred to (5.3). The revised fuel consumption cost of route i after removing customers m, $m + 1$ and n is given as follows:

$$\begin{aligned}C^{1}_{f}(i) &= E(R_i(0), R_i(1)) + E(R_i(1), R_i(2)) + \cdots + E(R_i(m-1), R_i(n)) \\ &\quad + E(R_i(n), R_i(m+2)) + \cdots + E(R_i(n-1), R_i(m)) \\ &\quad + E(R_i(m), R_i(m+1)) + E(R_i(m), R_i(n+1)) \\ &\quad + \cdots + E(R_i(nc(R_i)), R_i(0))\end{aligned}$$

$$= \sum_{a=0}^{nc(R_i)} \sum_{b=a+1}^{nc(R_i)+1} E(R_i'(a), R_i'(b))$$

We subtract the revised cost ($C_f^1(i)$) from the initial cost (5.3) to acquire the change in cost for the load-based intra-route (2,1) operator ($\Delta^f_{m^+n}(i)$). The formulation is then simplified in the following way.

$$\begin{aligned}\Delta^f_{m^+n}(i) &= C_f^0(i) - C_f^1(i) \\ &= \alpha_0 \cdot [\Delta^d_{m^+n}(i)] + \alpha_1 \cdot [I^f_{m^+n}(i)]\end{aligned}$$

where $I^f_{m^+n}(i)$ is the load-based component in the formula which is expressed in two conditions as follows:

(a) Case 1: $n > m$,

$$\begin{aligned}I^f_{m^+n}(i) = {} & L(R_i(m-1), R_i(m)) \cdot (d(R_i(m-1), R_i(m)) \\ & - d(R_i(m-1), R_i(n))) \\ & + L(R_i(m+1), R_i(m+2)) \cdot (d(R_i(m), R_i(m+1)) \\ & + d(R_i(m+1), R_i(m+2))) \\ & - (L(R_i(m-1), R_i(m)) - q(R_i(n))) \cdot d(R_i(n), R_i(m+2))) \\ & + (q(R_i(n)) - q(R_i(m)) - q(R_i(m+1))) \\ & \cdot \sum_{a=m+2}^{n-2} \sum_{b=a+1}^{n-1} d(R_i(a), R_i(b)) \\ & + L(R_i(n-1), R_i(n)) \cdot d(R_i(n-1), R_i(n)) \\ & - (L(R_i(n), R_i(n+1)) + q(R_i(m)) + q(R_i(m+1))) \\ & \cdot (d(R_i(n-1), R_i(m)) \\ & + L(R_i(n), R_i(n+1)) \cdot (d(R_i(n), R_i(n+1)) \\ & - d(R_i(m), R_i(m+1)) - d(R_i(m+1), R_i(n+1)))\end{aligned}$$

If $n = m + 2$, $I^f_{m^+n}(i)$ will be reduced to

$$\begin{aligned}I^f_{m^+n}(i) = {} & L(R_i(m-1), R_i(m)) \cdot (d(R_i(m-1), R_i(m)) \\ & - d(R_i(m-1), R_i(m+2))) \\ & + L(R_i(m+1), R_i(m+2)) \cdot (d(R_i(m), R_i(m+1)) \\ & + d(R_i(m+1), R_i(m+2)))\end{aligned}$$

$$
\begin{aligned}
&- (L(R_i(m+2), R_i(m+3)) + q(R_i(m)) + q(R_i(m+1))) \\
&\quad \cdot (d(R_i(m+2), R_i(m)) \\
&+ L(R_i(m+2), R_i(m+3)) \cdot (d(R_i(m+2), R_i(m+3)) \\
&\quad - d(R_i(m), R_i(m+1)) - d(R_i(m+1), R_i(m+3)))
\end{aligned}
$$

(b) Case 2: $n < m$,

$$
\begin{aligned}
I^f_{m^+n}(i) = {} & L(R_i(n-1), R_i(n)) \cdot (d(R_i(n-1), R_i(n)) \\
& - d(R_i(n-1), R_i(m))) \\
& + L(R_i(n), R_i(n+1)) \cdot d(R_i(n), R_i(n+1)) \\
& - (L(R_i(n-1), R_i(n)) - q(R_i(m)) - q(R_i(m+1))) \\
& \quad \cdot (d(R_i(m), R_i(m+1)) + d(R_i(m+1), R_i(n+1))) \\
& + (q(R_i(m)) + q(R_i(m+1)) - q(R_i(n))) \\
& \quad \cdot \sum_{a=n+1}^{m-2} \sum_{b=a+1}^{m-1} d(R_i(a), R_i(b)) \\
& + L(R_i(m-1), R_i(m)) \cdot d(R_i(m-1), R_i(m)) \\
& - (L(R_i(m+1), R_i(m+2)) + q(R_i(n))) \cdot d(R_i(m-1), R_i(n)) \\
& + L(R_i(m+1), R_i(m+2)) \cdot (d(R_i(m), R_i(m+1)) \\
& \quad + d(R_i(m+1), R_i(m+2)) - d(R_i(n), R_i(m+2)))
\end{aligned}
$$

If $n = m - 1$, $I^f_{m^+n}(i)$ will be reduced to

$$
\begin{aligned}
I^f_{m^+n}(i) = {} & L(R_i(m-2), R_i(m-1)) \cdot (d(R_i(m-2), R_i(m-1)) \\
& - d(R_i(m-2), R_i(m))) \\
& + L(R_i(m-1), R_i(m)) \cdot d(R_i(m-1), R_i(m)) \\
& - (L(R_i(m+1), R_i(m+2)) + q(R_i(m-1))) \cdot (d(R_i(m), R_i(m+1)) \\
& \quad + d(R_i(m+1), R_i(m-1))) \\
& + L(R_i(m+1), R_i(m+2)) \cdot (d(R_i(m), R_i(m+1)) \\
& \quad + d(R_i(m+1), R_i(m+2)) - d(R_i(m-1), R_i(m+2)))
\end{aligned}
$$

Illustrative example

Consider an example of a route $R(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0)$ comprising of nine customers as illustrated in Figure 5.5. This route has a length of 39.66 units and would consume as much fuel as 6.79 units. By executing the intra-route (2,1)

interchange move, it seems to be useful to swap a link of customers 3 and 4 with customer 7. The route would then become $R'(0, 1, 2, 7, 5, 6, 3, 4, 8, 9, 0)$ which has a total length of 37.93 units and uses a fuel consumption of 6.41 units. This interchange manages to reduce the route length by 1.73 units and fuel consumption by 0.38 units. Both the classical and the load-based (2,1) operators produced the same solution.

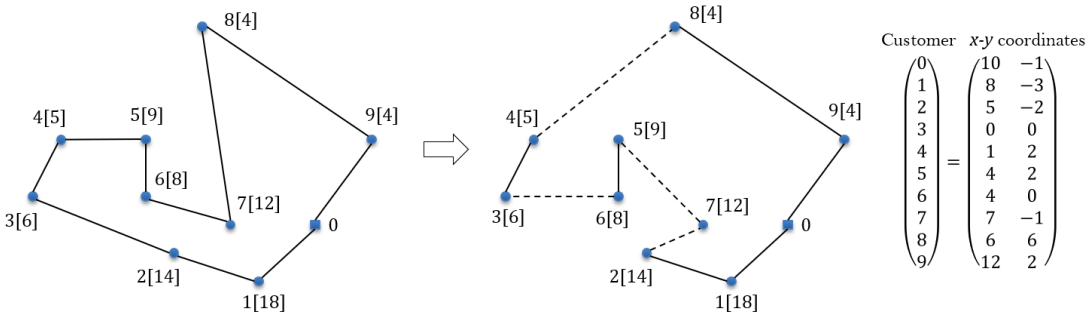

FIGURE 5.5: An example of (2,1) intra-route move

5.4.2 The inter-route (2,1)

For the procedure among the routes, the removal process in route i is referred to (4.3), (4.19) and (4.20). The cost of route j before and after the removal of customer n can be referred to (5.4) and (5.5). Thus, we can acquire the total removal cost ($\delta^{rd}_{m^+n}(ij)$) as follows:

$$\begin{aligned}\delta^{rd}_{m^+n}(ij) &= C^0_d(i) - C^1_d(i) + C^0_d(j) - C^1_d(j) \\ &= d(R_i(m-1), R_i(m)) + d(R_i(m), R_i(m+1)) \\ &\quad + d(R_i(m+1), R_i(m+2)) - d(R_i(m-1), R_i(m+2)) \\ &\quad + d(R_j(n-1), R_j(n)) + d(R_j(n), R_j(n+1)) \\ &\quad - d(R_j(n-1), R_j(n+1))\end{aligned}$$

The updated route cost after inserting customer n in route i is already given in (5.7). The updated route cost after inserting customers m and $m+1$ in route j is also referred to (4.26). Therefore, the insertion cost ($\delta^{id}_{knlm^+}(ij)$) is obtained as follows:

$$\delta^{id}_{knlm^+}(ij) = C^2_d(i) - C^1_d(i) + C^2_d(j) - C^1_d(j)$$

$$
\begin{aligned}
&= d(R'_i(k), R_j(n)) + d(R_j(n), R'_i(k+1)) \\
&\quad - d(R'_i(k), R'_i(k+1)) + d(R'_j(l), R_i(m)) \\
&\quad + d(R_i(m), R_i(m+1)) + d(R_i(m+1), R'_j(l+1)) \\
&\quad - d(R'_j(l), R'_j(l+1))
\end{aligned}
$$

The difference between the removal and the insertion cost will produce the change in cost for the classical inter-route (2,1) operator which is denoted by $\Delta^{d}_{knlm^{+}}(ij)$. The formulation is expressed as

$$
\begin{aligned}
\Delta^{d}_{knlm^{+}}(ij) &= \delta^{rd}_{m^{+}n}(ij) - \delta^{id}_{knlm^{+}}(ij) \\
&= d(R_i(m-1), R_i(m)) + d(R_i(m+1), R_i(m+2)) \\
&\quad - d(R_i(m-1), R_i(m+2)) + d(R_j(n-1), R_j(n)) \\
&\quad + d(R_j(n), R_j(n+1)) - d(R_j(n-1), R_j(n+1)) \\
&\quad - d(R'_i(k), R_j(n)) - d(R_j(n), R'_i(k+1)) \\
&\quad + d(R'_i(k), R'_i(k+1)) - d(R'_j(l), R_i(m)) \\
&\quad - d(R_i(m+1), R'_j(l+1)) + d(R'_j(l), R'_j(l+1))
\end{aligned} \tag{5.17}
$$

Effect of the load factor

The initial and the new cost for route i after removing a pair of customers m and $m+1$ in terms of fuel consumption can be referred to (4.8) and (4.22). Whilst the initial and the new cost for route j after removing customer n can be referred to (5.10) and (5.11). The removal cost ($\delta^{rf}_{m^{+}n}(ij)$) with regard to fuel consumption is obtained as follows:

$$
\begin{aligned}
\delta^{rf}_{m^{+}n}(ij) &= C^0_f(i) - C^1_f(i) + C^0_f(j) - C^1_f(j) \\
&= \alpha_0[d(R_i(m-1), R_i(m)) + d(R_i(m), R_i(m+1)) \\
&\quad + d(R_i(m+1), R_i(m+2)) - d(R_i(m-1), R_i(m+2)) \\
&\quad + d(R_j(n-1), R_j(n)) + d(R_j(n), R_j(n+1)) \\
&\quad - d(R_j(n-1), R_j(n+1))] \\
&\quad + \alpha_1[(q(R_i(m)) + q(R_i(m+1))) \cdot \sum_{a=0}^{m-2} \sum_{b=a+1}^{m-1} d(R_i(a), R_i(b)) \\
&\quad + L(R_i(m-1), R_i(m)) \cdot d(R_i(m-1), R_i(m)) \\
&\quad + L(R_i(m), R_i(m+1)) \cdot d(R_i(m), R_i(m+1)) \\
&\quad + L(R_i(m+1), R_i(m+2)) \cdot (d(R_i(m+1), R_i(m+2))
\end{aligned}
$$

$$
\begin{aligned}
&\quad - d(R_i(m-1), R_i(m+2)))\\
&+ q(R_j(n)) \cdot \sum_{a=0}^{n-2} \sum_{b=a+1}^{n-1} d(R_j(a), R_j(b))\\
&+ L(R_j(n-1), R_j(n)) \cdot d(R_j(n-1), R_j(n))\\
&+ L(R_j(n), R_j(n+1)) \cdot (d(R_j(n), R_j(n+1))\\
&\quad - d(R_j(n-1), R_j(n+1)))]
\end{aligned}
$$

Once customers m and $m+1$ are inserted in route j and customer n in route i, the fuel consumption cost for routes i and j are revised based on (5.13) and (4.29). The insertion cost ($\delta^{if}_{knlm^+}(ij)$) with regard to fuel consumption is calculated as follows:

$$
\begin{aligned}
\delta^{if}_{knlm^+}(ij) &= C^2_f(i) - C^1_f(i) + C^2_f(j) - C^1_f(j)\\
&= \alpha_0[d(R'_i(k), R_j(n)) + d(R_j(n), R'_i(k+1))\\
&\quad - d(R'_i(k), R'_i(k+1)) + d(R'_j(l), R_i(m))\\
&\quad + d(R_i(m), R_i(m+1)) + d(R_i(m+1), R'_j(l+1))\\
&\quad - d(R'_j(l), R'_j(l+1))]\\
&\quad + \alpha_1[q(R_j(n)) \sum_{a=0}^{k-1} \sum_{b=a+1}^{k} d(R'_i(a), R'_i(b))\\
&\quad + (L(R'_i(k), R'_i(k+1)) + q(R_j(n))) \cdot d(R'_i(k), R_j(n))\\
&\quad + L(R'_i(k), R'_i(k+1)) \cdot (d(R_j(n), R'_i(k+1))\\
&\quad\quad - d(R'_i(k), R'_i(k+1)))\\
&\quad + (q(R_i(m)) + q(R_i(m+1))) \cdot \sum_{a=0}^{l-1} \sum_{b=a+1}^{l} d(R'_j(a), R'_j(b))\\
&\quad + (L(R'_j(l), R'_j(l+1)) + q(R_i(m)) + q(R_i(m+1)))\\
&\quad\quad \cdot d(R'_j(l), R_i(m))\\
&\quad + (L(R'_j(l), R'_j(l+1)) + q(R_i(m+1))) \cdot d(R_i(m), R_i(m+1))\\
&\quad + L(R'_j(l), R'_j(l+1)) \cdot (d(R_i(m+1), R'_j(l+1))\\
&\quad\quad - d(R'_j(l), R'_j(l+1)))]
\end{aligned}
$$

The change in cost for the load-based inter-route (2,1) operator can be expressed as follows:

$$\Delta^f_{knlm^+}(ij) = \delta^{rf}_{m^+n}(ij) - \delta^{if}_{knlm^+}(ij)$$

$$
\begin{aligned}
&= \alpha_0[d(R_i(m-1), R_i(m)) + d(R_i(m+1), R_i(m+2)) \\
&\quad - d(R_i(m-1), R_i(m+2)) + d(R_j(n-1), R_j(n)) \\
&\quad + d(R_j(n), R_j(n+1)) - d(R_j(n-1), R_j(n+1)) \\
&\quad - d(R'_i(k), R_j(n)) - d(R_j(n), R'_i(k+1)) \\
&\quad + d(R'_i(k), R'_i(k+1)) - d(R'_j(l), R_i(m)) \\
&\quad - d(R_i(m+1), R'_j(l+1)) + d(R'_j(l), R'_j(l+1))] \\
&\quad + \alpha_1[(q(R_i(m)) + q(R_i(m+1))) \cdot (\sum_{a=0}^{m-2} \sum_{b=a+1}^{m-1} d(R_i(a), R_i(b)) \\
&\qquad - \sum_{c=0}^{l-1} \sum_{d=c+1}^{l} d(R'_j(c), R'_j(d))) \\
&\quad + L(R_i(m-1), R_i(m)) \cdot d(R_i(m-1), R_i(m)) \\
&\quad + L(R_i(m+1), R_i(m+2)) \cdot (d(R_i(m), R_i(m+1)) \\
&\qquad + d(R_i(m+1), R_i(m+2)) - d(R_i(m-1), R_i(m+2))) \\
&\quad + q(R_j(n)) \cdot (\sum_{a=0}^{n-2} \sum_{b=a+1}^{n-1} d(R_j(a), R_j(b)) \\
&\qquad - \sum_{c=0}^{k-1} \sum_{d=c+1}^{k} d(R'_i(c), R'_i(d))) \\
&\quad + L(R_j(n-1), R_j(n)) \cdot d(R_j(n-1), R_j(n)) \\
&\quad + L(R_j(n), R_j(n+1)) \cdot (d(R_j(n), R_j(n+1)) \\
&\qquad - d(R_j(n-1), R_j(n+1))) \\
&\quad - (L(R'_i(k), R'_i(k+1)) + q(R_j(n))) \cdot d(R'_i(k), R_j(n)) \\
&\quad - L(R'_i(k), R'_i(k+1)) \cdot (d(R_j(n), R'_i(k+1)) \\
&\qquad - d(R'_i(k), R'_i(k+1))) \\
&\quad - (L(R'_j(l), R'_j(l+1)) + q(R_i(m)) + q(R_i(m+1))) \\
&\qquad \cdot d(R'_j(l), R_i(m)) \\
&\quad - L(R'_j(l), R'_j(l+1)) \cdot (d(R_i(m), R_i(m+1)) \\
&\qquad + d(R_i(m+1), R'_j(l+1)) - d(R'_j(l), R'_j(l+1)))]
\end{aligned}
\tag{5.18}
$$

Illustrative example

Consider an example of a network consisting of nine customers which are initially assigned to two routes $R_1(0, 1, 2, 3, 4, 0)$ and $R_2(0, 5, 6, 7, 8, 9, 0)$ as depicted in Figure 5.6. R_1 and R_2 have a length of 24.26 and 23.07 units and consume 3.93

and 3.84 units of fuel, respectively. Interesting outcomes when implementing the (2,1) inter-route operator are observed.

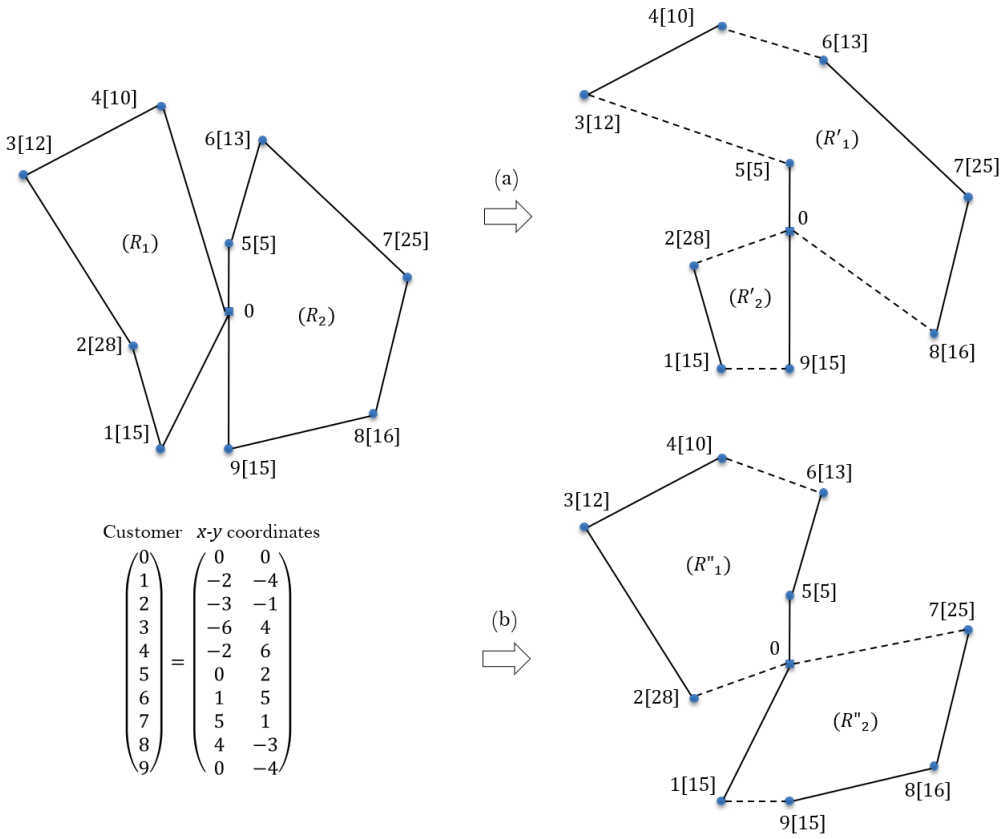

FIGURE 5.6: Sample network to show the move of inter-route (2,1) using (a) the classical and (b) the load-based operators

The classical (2,1) inter-route operator yields a set of solutions as in Figure 5.6(a). The best move from the operator is to choose the pair of customers 3 and 4 from R_1 to be exchanged with customer 9 from R_2. The pair of customers 3 and 4 is inserted next to customer 5 in R_2 while customer 9 is inserted in the first position after the departure from depot in R_1. After the reverse route evaluation, two renewed routes $R'_1(0, 2, 1, 9, 0)$ and $R'_2(0, 8, 7, 6, 4, 3, 5, 0)$ are formed with a length of 12.32 and 30.74 units and a fuel consumption of 1.99 and 5.09 units, respectively. This solution yields about 9% improvement in both the total length and the total amount fuel consumption.

On the other hand, the load-based (2,1) inter-route operator yields a different set of solutions. This can be illustrated in Figure 5.6(b). The search procedure has finalised to exchange customer 1 in R_1 with a pair of customers 5 and 6 in R_2. This leads to two updated routes which are $R''_1(0, 2, 3, 4, 6, 5, 0)$ and $R''_2(0, 1, 9, 8, 7, 0)$.

This solution has a total length of 21.79 and 19.82 units together with 3.56 and 3.28 units of fuel consumption for R_1'' and R_2'', respectively. Moreover, the solution gains about 12% improvement in both the total length and fuel consumption. Note that, the reverse route evaluation is conducted before commencing the operator. Thus, the initial solution for $R_2(0, 9, 8, 7, 6, 5, 0)$ was in the opposite route direction which has a fuel consumption of 3.82 units.

5.4.3 Analysis and computational results

We present the computational results for the (2,1) interchange operator in Table 5.5 for intra-route and inter-route cases. The average percent deviation is calculated to compare the solution obtained from using the load-based operators with the best and the worst case of the classical ones. The load-based operator is implemented on two initial solutions namely, the regular and the combined savings. In case the solution of the load-based operator is worst than the best-case of the classical one, a comparison with the worst-case is also performed to see the performance of the load-based with respect to this measure. The results could be reflect the upper and the lower bound when measuring the performance of the newly developed load-based operator.

For the intra-route case, all data sets seem to have remained constant when the load-based from the regular savings is compared with the best-case of the classical operator. The load-based operator from the combined savings performs better than the classical operator in general, except for the medium data set. Further illustration is shown in Figure 5.7.

For the inter-route case, it can be seen that the load-based (2,1) operator from the regular savings perform better than the classical one in the medium data set only. However, this operator when applied from the combined savings perform better mostly in the large data set only. This is indicated by the positive value of the average percent deviation. An illustration for the inter-route analysis of (2,1) interchange operator is given in Figure 5.8.

We also summarize the result of the load-based (2,1) interchange operator from the combined savings to examine the effect of the load factor in the formulation from the beginning. Table 5.6 reviews the results of LBC for the intra-route case. It can be observed that the small data set records the highest average percent deviation (2.30%) when $\alpha = 0.7$ while the lowest (-0.67%) could be found in the

TABLE 5.5: Average percent deviation (in %) of the load-based (2,1) against the classical (2,1) interchange operator

α	Data set	Intra-route				Inter-route			
		LBR	LWR	LBC	LWC	LBR	LWR	LBC	LWC
0.6	Small	0.00	1.13	2.15	3.26	-0.10	0.87	-0.31	0.66
	Medium	0.02	0.53	-0.67	-0.16	0.25	0.76	-0.79	-0.28
	Large	0.05	0.73	0.96	1.64	-0.17	0.51	0.98	1.65
0.7	Small	0.00	0.83	2.30	3.11	-0.10	0.61	-0.08	0.63
	Medium	0.02	0.39	-0.14	0.24	0.23	0.60	-0.53	-0.16
	Large	0.05	0.55	0.98	1.48	-0.20	0.30	0.42	0.91
0.8	Small	0.00	0.54	1.32	1.86	-0.09	0.37	-0.57	-0.10
	Medium	0.02	0.26	-0.31	-0.07	0.21	0.46	-0.55	-0.31
	Large	0.04	0.37	0.60	0.93	-0.23	0.10	0.01	0.33
0.9	Small	0.00	0.26	0.96	1.22	-0.09	0.13	-0.13	0.10
	Medium	0.02	0.14	-0.47	-0.35	0.20	0.32	-1.00	-0.88
	Large	0.04	0.21	0.59	0.75	-0.25	-0.10	0.00	0.16
1.0	Small	-0.01	-0.01	-0.01	-0.01	-0.09	-0.09	-0.09	-0.09
	Medium	0.02	0.02	0.02	0.02	0.18	0.18	0.18	0.18
	Large	0.04	0.04	0.04	0.04	-0.28	-0.28	-0.28	-0.28

medium data set when $\alpha = 0.6$. Out of 14, 11 instances in the small data set generate a positive percent deviation which accounts for the highest frequency of #best in each data set. Overall, the highest #best is when $\alpha = 0.6$ where 63% of all instances produce a positive percent deviation. On the contrary, the highest number of instances in the data set that has a negative percent deviation is obtained in the medium data set when $\alpha = 0.8$ (i.e. 12 out of 20 or 60% of the instances). The highest #worst (i.e. 21 out of 46 instances) for all instances could also be found at the same α value. Furthermore, the overall best percent deviation is acquired at 8.70% in the small data set and the worst percent deviation is at -6.21% in the medium data set. Both results are observed when $\alpha = 0.6$.

Table 5.7 reviews the results of LBC for the inter-route case. The average percent deviation is the highest at 0.98% although this is rather a small value. This corresponds to the highest count of positive percent deviation where it is produced by 10 of 12 instances in the large data set when $\alpha = 0.6$. On the other hand, the lowest average percent deviation is at -1.00% in the medium data set where 15 of 20 or 75% of the instances have a negative percent deviation when $\alpha = 0.9$. Additionally, the highest and the lowest percent deviations across all instances are found to be 12.25% and -8.79%, respectively.

TABLE 5.6: Summary table for LBC case of intra-route (2,1)

Data set		α				
		0.6	0.7	0.8	0.9	1.0
Small	AvgDev(%)	2.15	**2.30**	1.32	0.96	-0.01
(14 instances)	#best	**11**	**11**	9	7	0
	#worst	3	3	4	4	2
	Best(%)	**8.70**	8.25	8.28	8.36	0.00
	Worst(%)	-2.19	-1.54	-1.54	-3.92	-0.05
Medium	AvgDev(%)	**-0.67**	-0.14	-0.31	-0.47	0.02
(20 instances)	#best	9	10	8	10	4
	#worst	11	10	**12**	10	1
	Best(%)	2.98	1.74	1.76	3.46	0.16
	Worst(%)	**-6.21**	-3.65	-3.66	-4.60	-0.02
Large	AvgDev(%)	0.96	0.98	0.60	0.59	0.04
(12 instances)	#best	9	7	7	9	3
	#worst	3	5	5	3	1
	Best(%)	3.35	2.96	2.94	4.08	0.37
	Worst(%)	-1.96	-1.60	-2.72	-1.66	-0.01

TABLE 5.7: Summary table for LBC case of inter-route (2,1)

Data set		α				
		0.6	0.7	0.8	0.9	1.0
Small	AvgDev(%)	-0.31	-0.08	-0.57	-0.13	-0.09
(14 instances)	#best	5	7	6	7	4
	#worst	9	7	8	6	8
	Best(%)	11.69	11.88	12.07	**12.25**	10.68
	Worst(%)	**-8.79**	-8.78	-8.78	-8.77	-6.12
Medium	AvgDev(%)	-0.79	-0.53	-0.55	**-1.00**	0.18
(20 instances)	#best	7	7	8	5	11
	#worst	13	13	12	**15**	9
	Best(%)	3.05	3.03	2.13	3.75	2.91
	Worst(%)	-5.22	-4.20	-3.81	-4.55	-3.37
Large	AvgDev(%)	**0.98**	0.42	0.01	0.00	-0.28
(12 instances)	#best	**10**	8	7	5	5
	#worst	2	4	5	7	7
	Best(%)	4.88	2.83	2.32	5.19	1.22
	Worst(%)	-1.32	-1.90	-2.19	-2.59	-1.92

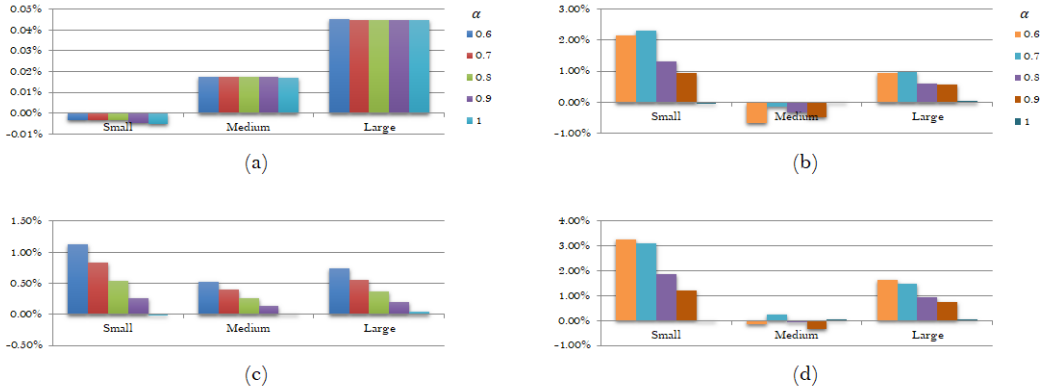

FIGURE 5.7: The average percent deviation of the load-based (2,1) exchange intra-route against the classical (2,1) exchange intra-route; (a) LBR, (b) LBC, (c) LWR, and (d) LWC

CPU times

The results in terms of the computational times of performing the (2,1) interchange operator is demonstrated in Table 5.8. The speed of processing the intra-route is almost negligible. All instances are completed roughly under 2 seconds. However, it can be seen for the inter-route, the average computational time for the large data set is drastically high compared to the small and medium data set. It can took almost up to 2 hours. This is recorded for an instance in the large data set. Thus, the overall average computational time for the inter-route is recorded at 1756, 1070 and 970 seconds for CR(2,1), LR(2,1) and LC(2,1) respectively.

TABLE 5.8: Average computational time for (2,1) (seconds)

Data set	Intra-route			Inter-route		
	CR(2,1)	LR(2,1)	LC(2,1)	CR(2,1)	LR(2,1)	LC(2,1)
Small	0.001	0.001	0.001	0.080	0.805	0.462
Medium	0.002	0.005	0.012	9.588	76.396	49.152
Large	0.016	0.086	0.082	6750.434	3975.881	3636.864
Overall	0.005	0.025	0.027	1765.176	1070.647	970.258

5.5 The (2,2) interchange operator

This operator searches two customers from one route to be exchanged with two customers from another route. In this study, we analyse the move involving two

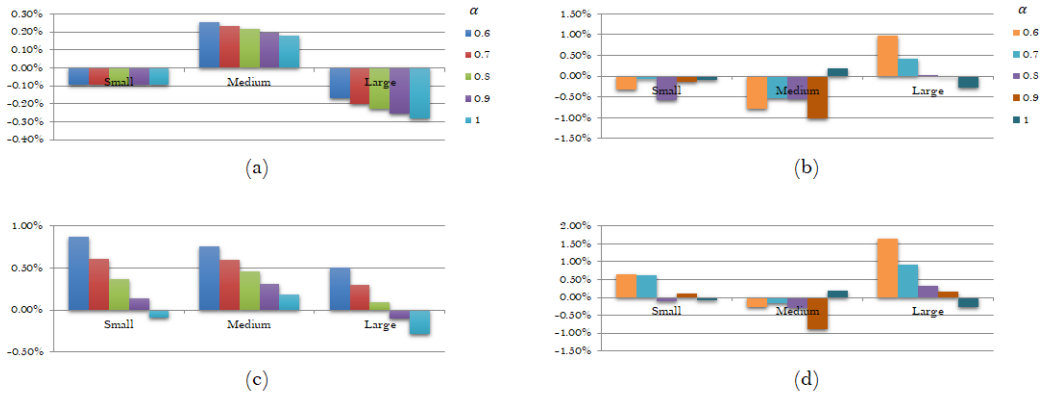

FIGURE 5.8: The average percent deviation of the load-based (2,1) exchange inter-route against the classical (2,1) exchange inter-route; (a) LBR, (b) LBC, (c) LWR, and (d) LWC

consecutive customers although the search can easily be extended to cater for any two customers. As in the previous operators, we divide the procedure of (2,2) interchange into two phases namely, the removal and the insertion process. This is done to explore the mathematical formulation when incorporating the load factor. These are based on the intra-route and the inter-route cases as presented below.

5.5.1 The intra-route (2,2)

The initial cost for route i is given in (5.1) and the updated cost of exchanging a pair of customers m and $m+1$ and a pair of customers n and $n+1$ is expressed as follows:

$$\begin{aligned} C_d^1(i) = {} & d(R_i(0), R_i(1)) + d(R_i(1), R_i(2)) + \cdots + d(R_i(m-1), R_i(n)) \\ & + d(R_i(n), R_i(n+1)) + d(R_i(n+1), R_i(m+2)) + \ldots \\ & + d(R_i(n-1), R_i(m)) + d(R_i(m), R_i(m+1)) \\ & + d(R_i(m+1), R_i(n+2)) + \cdots + d(R_i(nc(R_i)), R_i(0)) \\ = {} & \sum_{a=0}^{nc(R_i)} \sum_{b=a+1}^{nc(R_i)+1} d(R_i'(a), R_i'(b)) \end{aligned}$$

Therefore, we can find the change in cost for the classical intra-route (2,2) operator ($\Delta^{d}_{m^{+}n^{+}}(i)$) which is based on the swap or direct exchange move, as follows:

$$\begin{aligned}\Delta^{d}_{m^{+}n^{+}}(i) &= C^{0}_{d}(i) - C^{1}_{d}(i) \\ &= d(R_i(m-1), R_i(m)) + d(R_i(m+1), R_i(m+2)) \\ &\quad + d(R_i(n-1), R_i(n)) + d(R_i(n+1), R_i(n+2)) \\ &\quad - d(R_i(m-1), R_i(n)) - d(R_i(n+1), R_i(m+2)) \\ &\quad - d(R_i(n-1), R_i(m)) - d(R_i(m+1), R_i(n+2))\end{aligned} \tag{5.19}$$

If $n = m + 2$, after basic mathematical manipulation the $\Delta^{d}_{m^{+}n^{+}}(i)$ is marginally reduced to

$$\begin{aligned}\Delta^{d}_{m^{+}n^{+}}(i) &= d(R_i(m-1), R_i(m)) + d(R_i(m+1), R_i(m+2)) \\ &\quad + d(R_i(m+3), R_i(m+4)) - d(R_i(m-1), R_i(m+2)) \\ &\quad - d(R_i(m+3), R_i(m)) - d(R_i(m+1), R_i(m+4))\end{aligned}$$

Effect of the load factor

Instead of expressing the route cost in distance measurement, we introduced a new formulation to incorporate the load factor with the fuel consumption expressed in measuring the route cost. The initial cost $C^{0}_{f}(i)$ is given in (5.3) and the new cost $C^{1}_{f}(i)$ after exchanging a pair of customers m and $m+1$ and a pair of customers n and $n+1$ in terms of fuel consumption is shown below.

$$\begin{aligned}C^{1}_{f}(i) &= E(R_i(0), R_i(1)) + E(R_i(1), R_i(2)) + \cdots + E(R_i(m-1), R_i(n)) \\ &\quad + E(R_i(n), R_i(n+1)) + E(R_i(n), R_i(m+2)) + \ldots \\ &\quad + E(R_i(n-1), R_i(m)) + E(R_i(m), R_i(m+1)) \\ &\quad + E(R_i(m+1), R_i(n+2)) + \cdots + E(R_i(nc(R_i)), R_i(0)) \\ &= \sum_{a=0}^{nc(R_i)} \sum_{b=a+1}^{nc(R_i)+1} E(R'_i(a), R'_i(b))\end{aligned}$$

Hence, the change in cost for the load-based intra-route (2,2) operator, which is denoted as $\Delta^{f}_{m^{+}n^{+}}(i)$, is obtained as follows:

$$
\begin{aligned}
\Delta^{f}_{m^{+}n^{+}}(i) &= C^{0}_{f}(i) - C^{1}_{f}(i) \\
&= \alpha_0[d(R_i(m-1),R_i(m)) + d(R_i(m+1),R_i(m+2)) \\
&\quad + d(R_i(n-1),R_i(n)) + d(R_i(n+1),R_i(n+2)) \\
&\quad - d(R_i(m-1),R_i(n)) - d(R_i(n+1),R_i(m+2)) \\
&\quad - d(R_i(n-1),R_i(m)) - d(R_i(m+1),R_i(n+2))] \\
&\quad + \alpha_1[L(R_i(m-1),R_i(m)) \cdot (d(R_i(m-1),R_i(m)) \\
&\qquad - d(R_i(m-1),R_i(n))) \\
&\quad + L(R_i(m+1),R_i(m+2)) \cdot (d(R_i(m),R_i(m+1)) \\
&\qquad - d(R_i(m+1),R_i(m+2))) \\
&\quad - (L(R_i(m-1),R_i(m)) - q(R_i(n)) - q(R_i(n+1))) \\
&\qquad \cdot (d(R_i(n),R_i(n+1))) + d(R_i(n+1),R_i(m+2))) \\
&\quad + \begin{cases} (q(R_i(n)) + q(R_i(n+1)) - q(R_i(m)) \\ \quad - q(R_i(m+1))) \cdot \sum_{a=m+2}^{n-2} \sum_{b=a+1}^{n-1} d(R_i(a),R_i(b)) & \text{if } n-m>3, \\ 0 & \text{otherwise} \end{cases} \\
&\quad + L(R_i(n-1),R_i(n)) \cdot d(R_i(n-1),R_i(n)) \\
&\quad - (L(R_i(n+1),R_i(n+2)) + q(R_i(m)) + q(R_i(m+1))) \\
&\qquad \cdot (d(R_i(n-1),R_i(m)) \\
&\quad + L(R_i(n+1),R_i(n+2)) \cdot (d(R_i(n),R_i(n+1)) \\
&\qquad + d(R_i(n+1),R_i(n+2)) - d(R_i(m),R_i(m+1)) \\
&\qquad - d(R_i(m+1),R_i(n+2)))]
\end{aligned}
$$

For the special case when $n = m + 2$, the $\Delta^{f}_{m^{+}n^{+}}(i)$, after mathematical manipulation, is reduced and this is displayed as follows:

$$
\begin{aligned}
\Delta^{f}_{m^{+}n^{+}}(i) &= \alpha_0[d(R_i(m-1),R_i(m)) + d(R_i(m+1),R_i(m+2)) \\
&\qquad + d(R_i(m+3),R_i(m+4)) - d(R_i(m-1),R_i(m+2)) \\
&\qquad - d(R_i(m+3),R_i(m)) - d(R_i(m+1),R_i(m+4))] \\
&\quad + \alpha_1[L(R_i(m-1),R_i(m)) \cdot (d(R_i(m-1),R_i(m)) \\
&\qquad - d(R_i(m-1),R_i(m+2)))
\end{aligned}
$$

$$
\begin{aligned}
&+ L(R_i(m+1), R_i(m+2)) \cdot (d(R_i(m), R_i(m+1)) \\
&\quad + d(R_i(m+1), R_i(m+2))) \\
&- (L(R_i(m-1), R_i(m)) - q(R_i(m+2)) - q(R_i(m+3))) \\
&\quad \cdot (d(R_i(m+2), R_i(m+3)) + d(R_i(m+3), R_i(m))) \\
&+ L(R_i(m+3), R_i(m+4)) \cdot (d(R_i(m+2), R_i(m+3)) \\
&\quad + d(R_i(m+3), R_i(m+4)) - d(R_i(m), R_i(m+1)) \\
&\quad - d(R_i(m+1), R_i(m+4)))]
\end{aligned}
$$

Illustrative example

Consider an example consisting of eleven customers which are assigned to a route as illustrated in Figure 5.9. The initial route $R(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 0)$ has a length C_d of 61.52 units and fuel consumption C_e of 10.47 units. We implement the (2,2) interchange within the route. The best move is to swap a chain of customers 5 and 6 with a chain of customers 9 and 10. This would result in a renewed route $R'(0, 1, 2, 3, 4, 9, 10, 7, 8, 5, 6, 11, 0)$ with $C_d = 46.25$ and $C_e = 8.00$. The cost of the new route in terms of both total length and total fuel consumption is reduced about 24% from the initial cost. The intra-route classical and the load-based (2,2) interchange operators both produce the same route configuration of the solution.

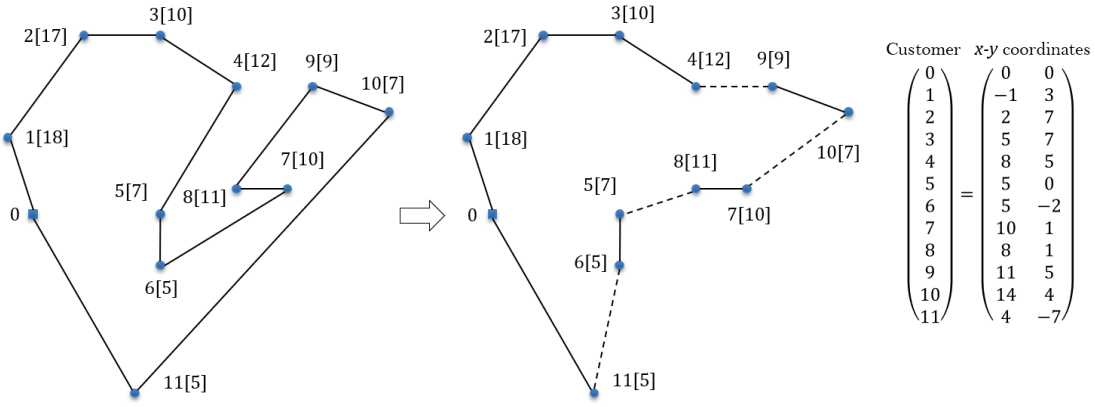

FIGURE 5.9: An example of (2,2) intra-route move

5.5.2 The inter-route (2,2)

The initial and the new cost of route i after removing customers m and $m+1$ may be referred to (4.3) and (4.19). The initial cost for route j is given in (5.4) and the new cost after the removal of customers n and $n+1$ is shown as follows:

$$\begin{aligned} C_d^1(j) &= d(R_j(0), R_j(1)) + d(R_j(1), R_j(2)) + \cdots + d(R_j(n-1), R_j(n+2)) \\ &\quad + \cdots + d(R_j(nc(R_j)-1), R_j(0)) \\ &= \sum_{a=0}^{nc(R_j)-1} \sum_{b=a+1}^{nc(R_j)} d(R'_j(a), R'_j(b)) \end{aligned}$$

The total removal cost $\delta^{rd}_{m^+n^+}(ij)$ is then calculated in the following.

$$\begin{aligned} \delta^{rd}_{m^+n^+}(ij) &= C_d^0(i) - C_d^1(i) + C_d^0(j) - C_d^1(j) \\ &= d(R_i(m-1), R_i(m)) + d(R_i(m), R_i(m+1)) \\ &\quad + d(R_i(m+1), R_i(m+2)) - d(R_i(m-1), R_i(m+2)) \\ &\quad + d(R_j(n-1), R_j(n)) + d(R_j(n), R_j(n+1)) \\ &\quad + d(R_j(n+1), R_j(n+2)) - d(R_j(n-1), R_j(n+2)) \end{aligned}$$

After inserting customers n and $n+1$ in route i, the route cost is revised to be

$$\begin{aligned} C_d^2(i) &= d(R'_i(0), R'_i(1)) + d(R'_i(1), R'_i(2)) + \cdots + d(R'_i(k), R_j(n)) \\ &\quad + d(R_j(n), R_j(n+1)) + d(R_j(n+1), R'_i(k+1)) + \ldots \\ &\quad + d(R'_i(nc(R_i)), R'_i(0)) \\ &= \sum_{a=0}^{nc(R_i)} \sum_{b=a+1}^{nc(R_i)+1} d(R''_i(a), R''_i(b)) \end{aligned}$$

The new route cost for route j after the insertion of customers m and $m+1$ is given in (4.26). Thus, the total insertion cost $\delta^{id}_{kn^+lm^+}(ij)$ is acquired as displayed below.

$$\begin{aligned} \delta^{id}_{kn^+lm^+}(ij) &= C_d^2(i) - C_d^1(i) + C_d^2(j) - C_d^1(j) \\ &= d(R'_i(k), R_j(n)) + d(R_j(n), R_j(n+1)) \\ &\quad + d(R_j(n+1), R'_i(k+1)) - d(R'_i(k), R'_i(k+1)) \\ &\quad + d(R'_j(l), R_i(m)) + d(R_i(m), R_i(m+1)) \end{aligned}$$

$$+ d(R_i(m+1), R'_j(l+1)) - d(R'_j(l), R'_j(l+1))$$

By subtracting the total insertion cost from the total removal cost, we can acquire the change in cost for the classical inter-route (2,2) operator which is denoted as $\Delta^d_{kn^+lm^+}(ij)$. The formulation is expressed as follows:

$$\begin{aligned}
\Delta^d_{kn^+lm^+}(ij) &= \delta^{rd}_{m^+n^+}(ij) - \delta^{id}_{kn^+lm^+}(ij) \\
&= d(R_i(m-1), R_i(m)) + d(R_i(m+1), R_i(m+2)) \\
&\quad - d(R_i(m-1), R_i(m+2)) + d(R_j(n-1), R_j(n)) \\
&\quad + d(R_j(n+1), R_j(n+2)) - d(R_j(n-1), R_j(n+2)) \\
&\quad - d(R'_i(k), R_j(n)) - d(R_j(n+1), R'_i(k+1)) \\
&\quad + d(R'_i(k), R'_i(k+1)) - d(R'_j(l), R_i(m)) \\
&\quad - d(R_i(m+1), R'_j(l+1)) + d(R'_j(l), R'_j(l+1))
\end{aligned} \tag{5.20}$$

Effect of the load factor

In terms of fuel consumption, the initial cost for routes i and j are given in (4.8) and (5.10). The updated cost of route i after removing customers m and $m+1$ can be referred to (4.22). For route j, the updated cost after removing customers n and $n+1$ is presented as follows:

$$\begin{aligned}
C^1_f(j) &= E(R_j(0), R_j(1)) + E(R_j(1), R_j(2)) + \cdots + E(R_j(n-1), R_j(n+2)) \\
&\quad + \cdots + E(R_j(nc(R_j)-1), R_j(0)) \\
&= \sum_{a=0}^{nc(R_j)-1} \sum_{b=a+1}^{nc(R_j)} E(R'_j(a), R'_j(b))
\end{aligned}$$

The total removal cost with regard to fuel consumption, which is represented by $\delta^{rf}_{m^+n^+}(ij)$, can be calculated as shown below.

$$\begin{aligned}
\delta^{rf}_{m^+n^+}(ij) &= C^0_f(i) - C^1_f(i) + C^0_f(j) - C^1_f(j) \\
&= \alpha_0[d(R_i(m-1), R_i(m)) + d(R_i(m), R_i(m+1)) \\
&\quad + d(R_i(m+1), R_i(m+2)) - d(R_i(m-1), R_i(m+2)) \\
&\quad + d(R_j(n-1), R_j(n)) + d(R_j(n), R_j(n+1)) \\
&\quad + d(R_j(n+1), R_j(n+2)) - d(R_j(n-1), R_j(n+2))]
\end{aligned}$$

$$
\begin{aligned}
&+ \alpha_1[(q(R_i(m)) + q(R_i(m+1))) \cdot (\sum_{a=0}^{m-2} \sum_{b=a+1}^{m-1} d(R_i(a), R_i(b)) \\
&\quad + L(R_i(m-1), R_i(m)) \cdot d(R_i(m-1), R_i(m)) \\
&\quad + L(R_i(m), R_i(m+1)) \cdot d(R_i(m), R_i(m+1)) \\
&\quad + L(R_i(m+1), R_i(m+2)) \cdot (d(R_i(m+1), R_i(m+2)) \\
&\qquad - d(R_i(m-1), R_i(m+2))) \\
&\quad + (q(R_j(n)) + q(R_j(n+1))) \cdot (\sum_{a=0}^{n-2} \sum_{b=a+1}^{n-1} d(R_j(a), R_j(b)) \\
&\quad + L(R_j(n-1), R_j(n)) \cdot d(R_j(n-1), R_j(n)) \\
&\quad + L(R_j(n), R_j(n+1)) \cdot d(R_j(n), R_j(n+1)) \\
&\quad + L(R_j(n+1), R_j(n+2)) \cdot (d(R_j(n+1), R_j(n+2)) \\
&\qquad - d(R_j(n-1), R_j(n+2)))]
\end{aligned}
$$

The insertion of customers n and $n+1$ in route i leads to a revised route cost which is expressed as

$$
\begin{aligned}
C_f^2(i) &= E(R_i'(0), R_i'(1)) + E(R_i'(1), R_i'(2)) + \cdots + E(R_i'(k), R_j(n)) \\
&\quad + E(R_j(n), R_j(n+1)) + E(R_j(n+1), R_i'(k+1)) + \ldots \\
&\quad + E(R_i'(nc(R_i)), R_i'(0)) \\
&= \sum_{a=0}^{nc(R_i)} \sum_{b=a+1}^{nc(R_i)+1} E(R_i''(a), R_i''(b))
\end{aligned}
$$

The revised cost for route j after inserting customers m and $m+1$ is given in (4.29). Subsequently, we can calculate the total insertion cost in terms of fuel consumption ($\delta^{if}_{kn^+lm^+}(ij)$) as follows:

$$
\begin{aligned}
\delta^{if}_{kn^+lm^+}(ij) &= C_f^2(i) - C_f^1(i) + C_f^2(j) - C_f^1(j) \\
&= \alpha_0[d(R_i'(k), R_j(n)) + d(R_j(n), R_j(n+1)) \\
&\quad + d(R_j(n+1), R_i'(k+1)) - d(R_i'(k), R_i'(k+1)) \\
&\quad + d(R_j'(l), R_i(m)) + d(R_i(m), R_i(m+1)) \\
&\quad + d(R_i(m+1), R_j'(l+1)) - d(R_j'(l), R_j'(l+1))] \\
&\quad + \alpha_1[(q(R_j(n)) + q(R_j(n+1))) \cdot \sum_{a=0}^{k-1} \sum_{b=a+1}^{k} d(R_i'(a), R_i'(b))) \\
&\quad + (L(R_i'(k), R_i'(k+1)) + q(R_j(n)) + q(R_j(n+1))) \\
&\qquad \cdot d(R_i'(k), R_j(n))
\end{aligned}
$$

$$
\begin{aligned}
&+ (L(R'_i(k), R'_i(k+1)) + q(R_j(n+1))) \cdot d(R_j(n), R_j(n+1)) \\
&+ L(R'_i(k), R'_i(k+1)) \cdot (d(R_j(n+1), R'_i(k+1)) \\
&\quad - d(R'_i(k), R'_i(k+1))) \\
&+ (q(R_i(m)) + q(R_i(m+1))) \cdot \sum_{c=0}^{l-1} \sum_{d=c+1}^{l} d(R'_j(c), R'_j(d))) \\
&+ (L(R'_j(l), R'_j(l+1)) + q(R_i(m)) + q(R_i(m+1))) \\
&\quad \cdot d(R'_j(l), R_i(m)) \\
&+ (L(R'_j(l), R'_j(l+1)) + q(R_i(m+1))) \cdot d(R_i(m), R_i(m+1)) \\
&+ L(R'_j(l), R'_j(l+1)) \cdot (d(R_i(m+1), R'_j(l+1)) \\
&\quad - d(R'_j(l), R'_j(l+1)))]
\end{aligned}
$$

The difference between the total removal cost ($\delta^{rf}_{m^+n^+}(ij)$) and the total insertion cost ($\delta^{if}_{kn^+lm^+}(ij)$) leads to the change in cost for the load-based inter-route (2,2) operator which is represented by $\Delta^{f}_{kn^+lm^+}(ij)$. This is demonstrated in the following lines.

$$
\begin{aligned}
\Delta^{f}_{kn^+lm^+}(ij) &= \delta^{rf}_{m^+n^+}(ij) - \delta^{if}_{kn^+lm^+}(ij) \\
&= \alpha_0[d(R_i(m-1), R_i(m)) + d(R_i(m+1), R_i(m+2)) \\
&\quad - d(R_i(m-1), R_i(m+2)) + d(R_j(n-1), R_j(n)) \\
&\quad + d(R_j(n+1), R_j(n+2)) - d(R_j(n-1), R_j(n+2)) \\
&\quad - d(R'_i(k), R_j(n)) - d(R_j(n+1), R'_i(k+1)) \\
&\quad + d(R'_i(k), R'_i(k+1)) - d(R'_j(l), R_i(m)) \\
&\quad - d(R_i(m+1), R'_j(l+1)) + d(R'_j(l), R'_j(l+1))] \\
&\quad + \alpha_1[(q(R_i(m)) + q(R_i(m+1))) \cdot (\sum_{a=0}^{m-2} \sum_{b=a+1}^{m-1} d(R_i(a), R_i(b)) \\
&\qquad - \sum_{c=0}^{l-1} \sum_{d=c+1}^{l} d(R'_j(c), R'_j(d))) \\
&\quad + L(R_i(m-1), R_i(m)) \cdot d(R_i(m-1), R_i(m)) \\
&\quad + L(R_i(m+1), R_i(m+2)) \cdot (d(R_i(m), R_i(m+1)) \\
&\qquad + d(R_i(m+1), R_i(m+2)) - d(R_i(m-1), R_i(m+2))) \\
&\quad + (q(R_j(n)) + q(R_j(n+1))) \cdot (\sum_{a=0}^{n-2} \sum_{b=a+1}^{n-1} d(R_j(a), R_j(b))
\end{aligned}
\tag{5.21}
$$

$$
\begin{aligned}
&\quad - \sum_{a=0}^{k-1} \sum_{b=a+1}^{k} d(R'_i(a), R'_i(b)))\\
&+ L(R_j(n-1), R_j(n)) \cdot d(R_j(n-1), R_j(n))\\
&+ L(R_j(n+1), R_j(n+2)) \cdot (d(R_j(n), R_j(n+1))\\
&\quad + d(R_j(n+1), R_j(n+2)) - d(R_j(n-1), R_j(n+2)))\\
&- (L(R'_i(k), R'_i(k+1)) + q(R_j(n)) + q(R_j(n+1)))\\
&\quad \cdot d(R'_i(k), R_j(n))\\
&- L(R'_i(k), R'_i(k+1)) \cdot (d(R_j(n), R_j(n+1))\\
&\quad + d(R_j(n+1), R'_i(k+1)) - d(R'_i(k), R'_i(k+1)))\\
&- (L(R'_j(l), R'_j(l+1)) + q(R_i(m)) + q(R_i(m+1)))\\
&\quad \cdot d(R'_j(l), R_i(m))\\
&- L(R'_j(l), R'_j(l+1)) \cdot (d(R_i(m), R_i(m+1))\\
&\quad + d(R_i(m+1), R'_j(l+1)) - d(R'_j(l), R'_j(l+1)))]
\end{aligned}
$$

Illustrative example

We present an example in Figure 5.10 to show the interchange of the (2,2) operator among the routes. Here we have two routes of which five customers are allocated in each route. Initially, the first route $R_1(0, 1, 2, 3, 4, 5, 0)$ has a length C_d of 24.78 units alongside the fuel consumption C_e of 4.17 units. The second route $R_2(0, 6, 7, 8, 9, 10, 0)$ has a length and fuel consumption of 25.28 and 4.29 units, respectively. The operator will search to move two consecutive customers from R_1 to R_2 and vice versa. This led to the interchange between a link of customers 1 and 2 with customers 6 and 7 which yielded two updated routes $R'_1(0, 3, 4, 5, 6, 7, 0)$ and $R'_2(0, 8, 9, 10, 1, 2, 0)$. The R'_1 and R'_2 each has C_d equal to 21.61 and 20.87 units and C_e of 3.71 and 3.53 units, respectively. This interchange gains 7.57 units in C_d and 1.22 units in C_e.

The same solution is obtained when we apply both the classical and the load-based (2,2) interchange operator. This means that the move is considered as appropriate in view of the length as well as the fuel consumption.

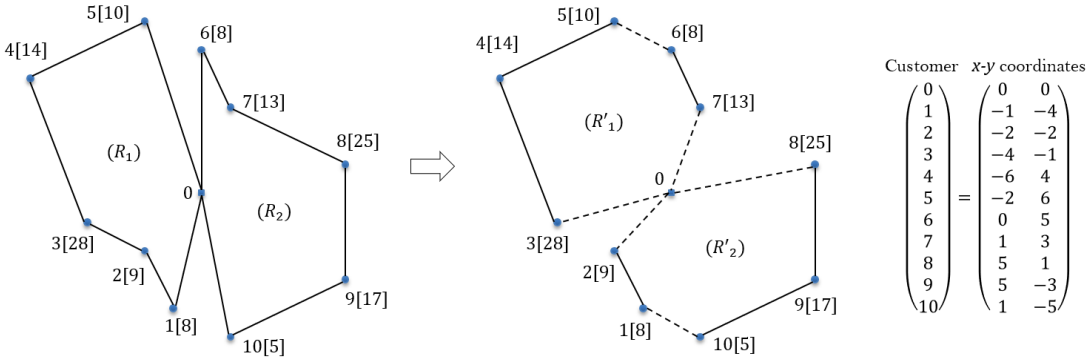

FIGURE 5.10: An example of (2,2) inter-route move

5.5.3 Analysis and computational results

The analysis for the (2,2) interchange operator is conducted similarly as the previous local search operators. We present the results in terms of the average percent deviation of the load-based operator against the classical operator for each data sets as in Table 5.9. Positive value of the percentage signifies an improvement of the solution using the newly developed load-based operator compared to the classical ones. The data is shown for five values of α which represent the weights applied to each objective as in (3.7). This table is customised to be in three decimals while others are all in two decimals point. This is especially because in the first column (i.e. intra-route LBR) all percentages are almost zero but in the chart it appears to be different values of 'zero' (see Figure 5.11(a)). The reason for this is because the solution from the load-based intra-route operator from the regular savings is as good as the solution from the best-case of the classical one. Therefore, the percent deviation between them is mostly zero.

Moreover, the load-based (2,2) intra-route operator from the combined savings generates mainly a better solution than the classical one. However, for some α values in the medium data set, this yields a negative average percent deviation. Figure 5.11 gives an illustration of the results for the intra-route (2,2) operator.

For the inter-route case, the best-case of the classical (2,2) operator seems to produce better solution than the load-based operator from the regular savings in the small and medium data sets, but not in the large data set. Compared to that, the load-based operator from the combined savings has a much better solution although some negative percentages could still be found, especially in the medium data set. This can be further illustrated in Figure 5.12. Note that the horizontal

TABLE 5.9: Average percent deviation (in %) of the load-based against the classical (2,2) interchange operator

α	Data set	Intra-route				Inter-route			
		LBR	LWR	LBC	LWC	LBR	LWR	LBC	LWC
0.6	Small	0.002	1.142	2.149	3.263	-0.144	0.998	1.709	2.828
	Medium	0.000	0.512	-0.622	-0.105	-0.203	0.337	-1.031	-0.484
	Large	0.000	0.692	0.932	1.617	0.210	0.875	1.043	1.701
0.7	Small	0.001	0.840	2.335	3.153	-0.156	0.683	1.887	2.707
	Medium	0.000	0.376	-0.133	0.243	-0.201	0.196	-0.638	-0.239
	Large	0.00	0.507	0.979	1.481	0.196	0.684	1.040	1.523
0.8	Small	0.000	0.547	1.347	1.886	-0.165	0.384	1.374	1.914
	Medium	0.000	0.246	-0.349	-0.102	-0.198	0.061	-0.727	-0.466
	Large	0.000	0.330	0.619	0.947	0.183	0.501	0.621	0.937
0.9	Small	-0.001	0.267	1.076	1.340	-0.173	0.096	0.768	1.034
	Medium	0.000	0.120	-0.514	-0.393	-0.196	-0.069	-0.772	-0.645
	Large	0.000	0.162	0.594	0.754	0.170	0.326	0.436	0.591
1.0	Small	-0.002	-0.002	-0.002	-0.002	-0.183	-0.183	-0.183	-0.183
	Medium	0.000	0.000	0.000	0.000	-0.194	-0.194	-0.194	-0.194
	Large	0.000	0.000	0.000	0.000	0.158	0.158	0.158	0.158

values seem to be the same when $\alpha = 1.0$ as this implies that the load factor is not relevant in the search process.

Table 5.10 summarizes the result of LBC for the case of intra-route. It can be shown in the small data set there are a few observations that is worth highlighted. First, at $\alpha = 0.7$ the highest average percent deviation is found to be 2.34%. Second, there are 11 of 14 (or about 79%) instances that has a positive percent deviation and this accounts for the highest #best. Third, the highest percent deviation over all instances is recorded at 8.64% when $\alpha = 0.9$. Apart from that, we can also observe the lowest average deviation percentage at -0.62% as well as the lowest value at -6.11% in the medium data set when $\alpha = 0.6$. The count of negative percent deviation is found to be the highest in the same data set in which 12 of 20 instances are found when $\alpha = 0.6$ and 0.8.

Furthermore, a summary result of LBC for the inter-route case is presented in Table 5.11. It is shown that the average percent deviation is at the highest of only 1.89% and the lowest at -1.03%. The largest #best for each data set is obtained in the large data set when $\alpha = 0.6$. For the overall result, the largest #best is obtained when $\alpha = 0.7$ where 50% of all instances produced a positive percentage. Whilst for #worst, the largest count for each data set is acquired when $\alpha = 0.6$ in the medium data set (16 out of 20 instances). Here, for the overall result, the

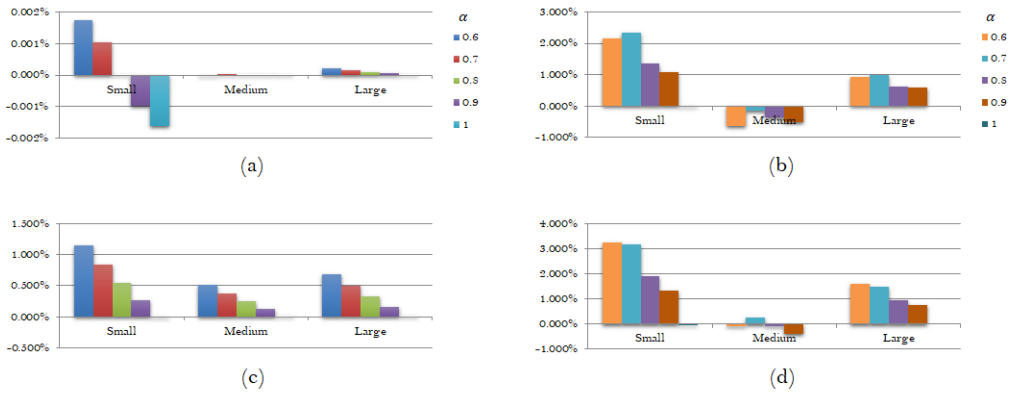

FIGURE 5.11: The average percent deviation of the load-based against the classical (2,2) interchange for the intra-route case; (a) LBR, (b) LBC, (c) LWR, and (d) LWC

largest count is acquired when $\alpha = 0.9$ where 29 out of 46 instances have a negative percent deviation. In addition, the best and the worst percent deviations for all instances are 8.64% and -5.89% in the small and medium data sets, respectively.

CPU times

The average computational time to find the solution using the (2,2) interchange operator is shown in Table 5.12. The data is shown for the small, medium and large data set as well as overall inclusive of 46 instances for CR(2,2), LR(2,2) and LC(2,2) for the intra-route and the inter-route cases. The average computational time for all instances of the intra-route case is found to be 5, 7 and 8 milliseconds together with the inter-route case which is 148, 182 and 229 seconds, respectively for CR(2,2), LR(2,2) and LC(2,2). It can be noted that the average computational time for the large data set is noticeably higher than in the medium and small data sets. This signifies that the search process is more difficult and complex in those instances from the large data set compared to the other two sets.

5.6 Summary

In this chapter, we examine three advanced local search operators namely, (1,1), (2,1) and (2,2) interchange operators. These operators are executed for both the intra-route and the inter-route cases. We adopted a swap move for the intra-route

TABLE 5.10: Summary table for LBC case of intra-route (2,2)

Data set		α				
		0.6	0.7	0.8	0.9	1.0
Small	AvgDev(%)	2.15	**2.34**	1.35	1.08	0.00
(14 instances)	#best	**11**	**11**	9	6	0
	#worst	3	3	4	6	1
	Best(%)	8.30	8.42	8.52	**8.64**	0.00
	Worst(%)	-2.46	-1.51	-1.50	-3.90	-0.02
Medium	AvgDev(%)	**-0.62**	-0.13	-0.35	-0.51	0.00
(20 instances)	#best	8	10	8	10	0
	#worst	**12**	10	**12**	10	0
	Best(%)	2.85	1.69	1.63	3.03	0.00
	Worst(%)	**-6.11**	-3.34	-3.35	-4.34	0.00
Large	AvgDev(%)	0.93	0.98	0.62	0.59	0.00
(12 instances)	#best	9	7	7	9	0
	#worst	3	5	5	3	0
	Best(%)	3.40	2.58	3.04	4.30	0.00
	Worst(%)	-2.27	-1.61	-2.97	-1.67	0.00

TABLE 5.11: Summary table for LBC case of inter-route (2,2)

Data set		α				
		0.6	0.7	0.8	0.9	1.0
Small	AvgDev(%)	1.71	**1.89**	1.37	0.77	-0.18
(14 instances)	#best	9	9	6	5	4
	#worst	5	5	8	8	6
	Best(%)	8.30	8.42	8.52	**8.64**	0.81
	Worst(%)	-1.76	-2.41	-2.41	-2.41	-2.55
Medium	AvgDev(%)	**-1.03**	-0.64	-0.73	-0.77	-0.19
(20 instances)	#best	4	6	6	5	8
	#worst	**16**	14	14	15	11
	Best(%)	3.24	2.25	2.10	3.88	0.50
	Worst(%)	**-5.89**	-3.95	-3.67	-5.18	-1.73
Large	AvgDev(%)	1.04	1.04	0.62	0.44	0.16
(12 instances)	#best	**9**	8	8	6	7
	#worst	3	4	4	6	4
	Best(%)	2.97	3.86	3.90	3.02	1.96
	Worst(%)	-2.17	-2.21	-2.88	-2.29	-0.91

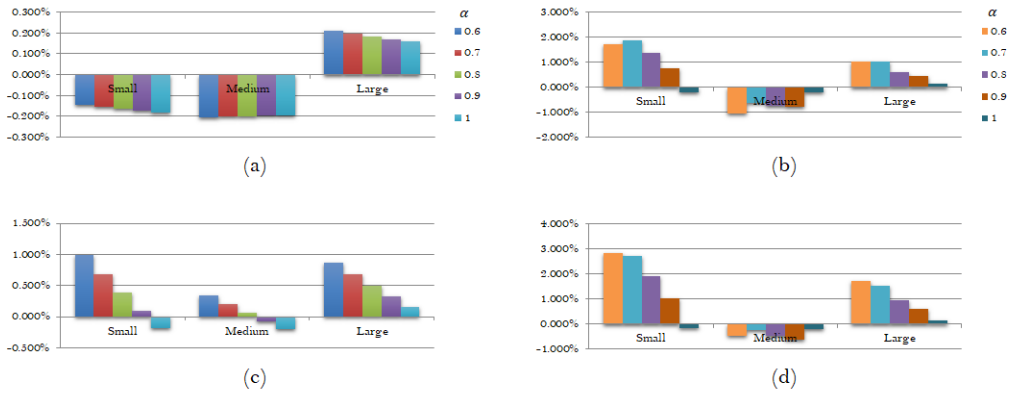

FIGURE 5.12: The average percent deviation of the load-based (2,2) exchange inter-route against the classical (2,2) exchange inter-route; (a) LBR, (b) LBC, (c) LWR, and
(d) LWC

TABLE 5.12: Average computational time for (2,2) (seconds)

Data set	Intra-route			Inter-route		
	CR(2,2)	LR(2,2)	LC(2,2)	CR(2,2)	LR(2,2)	LC(2,2)
Small	0.000	0.000	0.001	0.015	0.129	0.101
Medium	0.002	0.002	0.003	4.094	46.632	34.387
Large	0.013	0.022	0.024	561.048	622.481	820.804
Overall	0.005	0.007	0.008	148.145	182.700	229.105

case and the best exchange move for the inter-route case. We proposed new mathematical formulations that embed the load factor. We also presented an illustrative example for each case of the intra-route and the inter-route of the operators. To observe the performance of these newly developed operators, we conducted an extensive computational test using the 46 well-known VRP instances. We compared the solution of the load-based operators to the solution of the standard operators. Interesting results relating to its solution quality and computational time are obtained. In the next chapter, we assess the performance of all operators when used together. This is called out as part of the composite and multi-level heuristics.

Chapter 6

Incorporation of the load factor in composite and multi-level heuristics

6.1 Introduction

In this chapter, we examine the whole performance of the operators described in the previous chapters, 4 and 5. We perform the analysis using two powerful heuristics namely, the composite and the multi-level heuristics. We first provide a brief description of each of the two methods followed by the overall experimental design. The data sets used in the previous chapters are also used here to asses the performance of these methods where the load factor is attached. Additionally, we conduct a learning mechanism within the search for both heuristics. Our computational experiments over five runs reveal interesting results.

6.2 Integrated heurictics

Based on the computational results in the previous chapters, it can be noted that it may be inadequate to use just one single local search operator in generating better quality solutions. The solution could be enhanced by integrating the use of multiple operators in the search. This is performed by the composite and multi-level heuristics which will be explained in the next two subsections.

For clarity of the presentation, we denote the heuristic methods h as OI(h) which represents the use of the classical (O=C) and the load-based (O=L) operators from the initial solution using the regular (I=R) and the combined (I=C) savings.

Effect of the reverse route

In this integrated search, the evaluation of the reverse route direction schemes, as given in Chapter 3, is performed differently for the classical and the load-based operators. While using the classical operators, the evaluation scheme is applied just before the final solution is recorded. The scheme is used in the beginning once the initial solution is obtained from the savings method with the presence of the load-based operators.

6.2.1 Composite heuristic

Like the local search techniques, the composite heuristic starts with an initial solution. This sort of method is performed using a given improvement scheme also known as local search, to improve the solution. The search stops when no further improvement can be found. Such a solution is considered to be a local minimum with regards to the chosen scheme. To avoid getting trapped in this local minimum, the composite method directs the search by continuing the procedure sequentially with one or more other local searches. The idea is to create preferably as much alteration to the solution as possible so long as the total computing time is within a reasonable amount.

We adapt the load factor to the composite heuristic by applying the load-based techniques provided in the previous chapters. The heuristic we put forward consists of two parts. In the first part, a starting solution for CVRP is found. This is conducted as follows: Initially, the CVRP is solved using the savings method as described in Chapter 3. The regular savings method and our modified savings, namely the combined savings are chosen in obtaining the starting solution. Then, the 2-opt is applied to each route from the savings solution leading to our initial solution.

In the second part, a composite heuristic which attempts to improve on the solution found by applying a certain improvement scheme is introduced. The improvement schemes or formally known as the local search operators in the previous chapter are utilised here. The sequential order in which the operators are used

can be critical. As a rule, simple ones are adopted first. In our case the order is as follows: Starting with the (1,0) operator, followed by the (1,1) operator, (2,0) operator, (2,1) operator and lastly the (2,2) operator. Within each operator, both the intra-route and the inter-route cases are implemented one after another.

In addition to the above practice, we introduced two ways of carrying out this heuristic to include the 2-opt operator within the procedure besides using it to improve our starting solution.
(i) Firstly, we apply the 2-opt at the termination step before recording the final solution.
(ii) Secondly, we apply the 2-opt within the refinement scheme whenever an improvement is recorded.
It is worth noting that it would be good if the 2-opt could be applied only to the route(s) that underwent the modification after the refinement module is implemented. This can lead to a reduced computing time. Though this is not implemented here, it is interesting to pursue this issue in the future.

For simplicity, the first variant is represented by C1 and the latter by C2. The main steps of this algorithm are presented in Figures 6.1 and 6.2. The algorithm for C2 is similar to C1 with the exception of some of the steps as illustrated in the figure. It can be noted in the algorithm that the evaluation of the reverse route direction is employed differently when the classical and the load-based operators are used in the method. Also note, as all operators are finite therefore both C1 and C2 will systematically halt.

6.2.2 Multi-level heuristic

The multi-level heuristic is similar in principle with the composite heuristic. Yet, there are two plain distinctions between them. Firstly, in simple composite heuristic, during each step, an improvement scheme or the local search is applied in a small number, whereas here, in each level, a set of the local searches or a composite heuristic itself can be applied. Secondly, the composite heuristic continues to the next improvement scheme in sequence irrespective whether an improvement is found. However, for the multi-level heuristic, once the solution structure is altered, it is appropriate to direct the search and return back to the first level where it is usually calibrated with the simplest and a relatively faster scheme. Otherwise, the search will continue to the next level in sequence until there is no

further improvement. This approach is similar to variable neighbourhood descent (VND) as described in Salhi (2017).

In this study, we also adapt the load factor to the multi-level heuristic by incorporating the load-based techniques, as given in the previous chapters. The sequential order of the improvement schemes is similar to the one used in the composite method. From Level 1 to 5, the implementation of the operators is in the order as follows: the (1,0), (1,1), (2,0), (2,1) and (2,2) operators. Both the intra-route and the inter-route cases are activated consecutively in each level.

Besides improving the initial solution, we carry out two ways of featuring the 2-opt operator within the procedure. This is also similar to the one in the composite heuristic. Here, the first variant which is applied at the termination stage is represented by M1. The second one which is applied within each level once the solution is improved is called M2. The main steps of the algorithm for the multi-level heuristic is demonstrated in Figures 6.3 and 6.4. The algorithm for M2 is similar to M1 except for some of the steps in Level 1, 2, 5 and 6. Note that, here the reverse route evaluation scheme is applied at different levels when we use the classical or the load-based operators. Similar to C1 and C2, all operators used here are finite, hence M1 and M2 will systematically stop.

6.2.3 Analysis and computational result for the composite and multi-level heuristics

The experiments are designed as previewed in Figure 6.5. Two variants of incorporating the 2-opt operator in the search process are created for the composite and multi-level methods. The initial solution is obtained from the regular and the combined savings with the 2-opt applied afterwards. A set of the classical and the load-based operators are utilised to improve the solution using the composite and multi-level methods. This rises to twelve methods indexed by $k = 1, 2, \ldots, 12$. The computational tests are undertaken using the 46 well-known VRP instances as described in Figure 3.2 from Chapter 3. Five values of the weights, α are used in the tests. The α is set from 0.6 to 1.0 with an increment of 0.1.

For a given α, the objective function of the solution, Z (3.7) from one method is compared to the other method for a particular instance. The minimum Z value among twelve methods is recorded as the best Z. To show how much the solution

differs from each method with the best Z, we calculate the percent deviation as given in (3.8). Here, Z refers to the best Z and Z_k is the Z value obtained from method, k with $k \in \{1, 2, \ldots, 12\}$.

Solution quality

We present a sample of the computational results of the composite and multi-level heuristics when $\alpha = 0.6$. Table 6.1 summarizes the result where it shows the average percent deviation for the small, medium and large data sets as well as for the overall instances. It also shows the count of when the method k recorded the best Z. This is referred to as the total best Z. It can be observed that LC(C2) in the large data set produced the solution with the least average percent deviation which is at 0.24%. This would mean that most of the instances in the large data set produced a solution with the best Z or near to it. It can also be noted that LC(M2) recorded the highest count where 12 out of 46 or 26% of the instances are denoted as the best Z. Based on these observations, both methods LC(C2) and LC(M2) are considered for further exploration when the learning mechanism is introduced. This is described later in the next section.

TABLE 6.1: Summary table for the average percent deviation against the best Z (in %) and the total best Z of the composite and multi-level heuristics for $\alpha = 0.6$

Data set	CR(C1)	LR(C1)	LC(C1)	CR(C2)	LR(C2)	LC(C2)	CR(M1)	LR(M1)	LC(M1)	CR(M2)	LR(M2)	LC(M2)
Small	2.36	1.72	1.58	2.10	**1.53**	1.97	2.06	1.68	1.65	2.05	**1.53**	2.00
Medium	0.84	0.76	1.28	0.87	**0.75**	1.21	0.96	0.91	1.23	0.93	0.98	1.22
Large	2.43	1.56	0.34	2.28	1.18	**0.24**	2.02	1.41	0.51	1.95	1.58	0.25
Overall	1.72	1.26	1.12	1.61	**1.10**	1.19	1.57	1.28	1.17	1.54	1.30	1.20
Total Best Z	7	6	9	7	8	9	6	9	5	6	4	**12**

Composite heuristic (C1)

Step 0: (a) Generate an initial solution from the savings method and apply 2-opt.
(b) Optional: Evaluate the reverse route direction if using the *load-based* operators.
(c) Set the initial cost as C_0, $cycle \leftarrow 0$, #operators $p \leftarrow 5$, $opmax \leftarrow p$.

Step 1: (a) Apply the (1,0) operator, begin with the intra-route method and followed by the inter-route.
(b) Calculate the cost C_1 and the change $\delta_1 = C_0 - C_1$.
(c) If $\delta_1 > 0$, then
(i) set $cycle \leftarrow 0$
(ii) $C_0 \leftarrow C_1$
(d) Else (no change),
(i) $cycle \leftarrow cycle + 1$.
(ii) If $cycle = opmax$, then go to Step 6.

Step 2: Apply the (1,1) operator, repeat the procedure as in Step 1(b) - 1(d).

Step 3: Apply the (2,0) operator, repeat the procedure as in Step 1(b) - 1(d).

Step 4: Apply the (2,1) operator, repeat the procedure as in Step 1(b) - 1(d).

Step 5: Apply the (2,2) operator, repeat the procedure as in Step 1(b) - 1(d).

Step 6: (a) Optional: Evaluate reverse route direction if using the *classical* operators.
(b) Apply 2-opt when finishing with the operators in the previous step.
(c) Record the final solution and stop

FIGURE 6.1: Algorithm of the composite heuristic for C1

Composite heuristic (C2)

Step 0: *Same as* Step 0 *in C1.*

Step 1: *Same as* Step 1 *in C1, except for, in* (c)(ii) *the procedure is exchanged as follows:*

(c) (ii) Apply 2-opt and recalculate the cost C_2
and the change $\delta_2 = C_1 - C_2$.
If $\delta_2 > 0$, then set $C_0 \leftarrow C_2$,
else
$C_0 \leftarrow C_1$
$opmax \leftarrow p - 1$

Step 2: *Unchanged as in C1.*

Step 3: *Unchanged as in C1.*

Step 4: *Unchanged as in C1.*

Step 5: *Unchanged as in C1.*

Step 6: *Same as* Step 6 *in C1, except for* (b) *which does not applies in this variant*

FIGURE 6.2: Algorithm of the composite heuristic for C2

Multi-level heuristic (M1)

Level 0: (a) Generate an initial solution from the savings method and apply 2-opt.
(b) Optional: Evaluate the reverse route direction if using the *load-based* operators.
(c) Set the initial cost as C_0, #operators $p = 5$, *opcount* is number of p that has been used.

Level 1: (a) Set $opcount \leftarrow 0$.
(b) Apply the (1,0) operator, begin with the intra-route method and followed by the inter-route.
(c) Calculate the cost C_1 and the change $\delta_1 = C_0 - C_1$.
(d) If $\delta_1 > 0$, then
    (i) set $C_0 \leftarrow C_1$.
(e) Set $opcount \leftarrow 1$.

Level 2: (a) Apply the (1,1) operator, begin with the intra-route method and followed by the inter-route.
(b) Calculate the cost C_1 and the change $\delta_1 = C_0 - C_1$.
(c) If $\delta_1 > 0$, then
    (i) set $C_0 \leftarrow C_1$.
    (ii) Go back to Level 1.
(d) Else (no change),
    (i) set $opcount \leftarrow opcount + 1$.

Level 3: Apply the (2,0) operator, repeat the procedure as in Level 2(b) - 2(d).

Level 4: Apply the (2,1) operator, repeat the procedure as in Level 2(b) - 2(d).

Level 5: (a) Apply the (2,2) operator, begin with the intra-route method and followed by the inter-route method.
(b) Calculate the cost C_1 and the change $\delta_1 = C_0 - C_1$.
(c) If $\delta_1 > 0$, then
    (i) set $C_0 \leftarrow C_1$.
    (ii) Go back to Level 1.
(d) Else (no change),
    (i) if $opcount = op - 1$, then go to Level 6.

Level 6: (a) Optional: Evaluate reverse route direction if using the *classical* operator.
(b) Apply 2-opt when finishing with the operators in the previous level.
(c) Record the final solution.

FIGURE 6.3: Algorithm of the multi-level heuristic for M1

Multi-level heuristic (M2)

Level 0: *Same as* Level 0 *in M1.*

Level 1: *Same as* Level 1 *in M1, except for, in* (d)(i) *the step is replaced with*
(d) (i) Apply 2-opt and recalculate the cost C_2
and the change $\delta_2 = C_1 - C_2$.
If $\delta_2 > 0$, then set $C_0 \leftarrow C_2$,
else $C_0 \leftarrow C_1$.

Level 2: *Same as* Level 2 *in M1, except for, in* (c)(i) *the step is replaced with*
(c) (i) Apply 2-opt and recalculate the cost C_2
and the change $\delta_2 = C_1 - C_2$.
If $\delta_2 > 0$, then set $C_0 \leftarrow C_2$,
else $C_0 \leftarrow C_1$.

Level 3: *Unchanged as in M1.*

Level 4: *Unchanged as in M1.*

Level 5: *Same as* Level 5 *in M1, except for, in* (c)(i) *the step is replaced with*
(c) (i) Apply 2-opt and recalculate the cost C_2
and the change $\delta_2 = C_1 - C_2$.
If $\delta_2 > 0$, then set $C_0 \leftarrow C_2$,
else $C_0 \leftarrow C_1$.

Level 6: *Same as* Level 6 *in M1, except for* (b) *which does not applies in this variant.*

FIGURE 6.4: Algorithm of the multi-level heuristic for M2

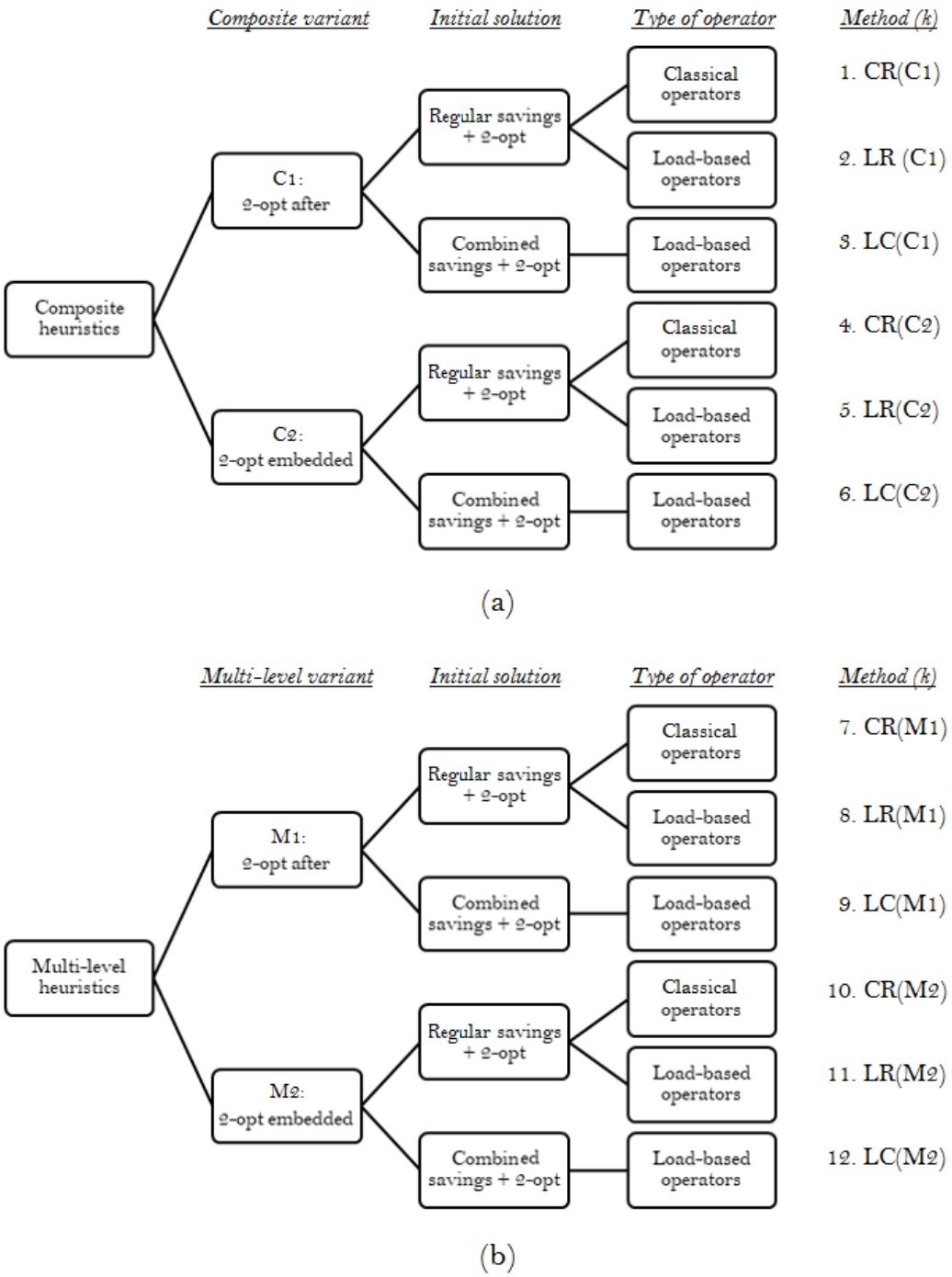

FIGURE 6.5: Test experiments using the (a) composite and (b) multi-level heuristics

TABLE 6.2: Detailed percent deviation (in %) against the best Z of the composite and multi-level heuristics for $\alpha = 0.6$

# (n)	Best Z	CR(C1)	LR(C1)	LC(C1)	CR(C2)	LR(C2)	LC(C2)	CR(M1)	LR(M1)	LC(M1)	CR(M2)	LR(M2)	LC(M2)
Small													
F1 (44)	112.14	3.44	0.47	7.08	0.00	2.03	7.45	0.00	0.47	7.08	0.00	2.03	7.45
F2 (71)	41.62	5.33	2.69	0.26	5.33	2.69	0.00	5.33	2.69	0.26	5.33	2.69	0.00
F3 (134)	190.96	0.21	0.00	0.67	0.06	0.20	0.67	0.21	0.00	0.67	0.06	0.20	0.67
C1 (21)	58.23	0.00	2.70	2.70	0.00	0.00	0.00	0.00	2.70	2.70	0.00	0.00	0.00
C2 (22)	99.71	1.52	0.00	3.19	1.52	0.00	1.93	1.52	0.00	3.19	1.52	0.00	1.93
C3 (29)	78.54	0.33	0.00	0.00	0.33	0.00	0.00	0.33	0.00	0.00	0.33	0.00	0.00
C4 (32)	179.16	0.11	0.81	0.00	0.11	0.81	0.00	0.11	0.81	0.00	0.11	0.81	0.00
C5 (50)	87.06	5.81	2.02	0.22	5.81	0.00	2.77	5.44	1.99	0.22	5.44	0.92	2.77
C6 (75)	135.82	0.47	0.00	1.74	0.47	0.62	1.74	0.47	0.62	1.74	0.47	0.62	1.74
C7 (100)	138.27	4.12	6.62	0.00	4.12	6.62	8.20	6.49	6.62	3.72	6.49	6.62	8.20
C8 (150)	172.99	7.09	6.23	3.27	7.09	5.88	0.00	6.20	5.94	0.38	6.20	5.86	0.21
C9 (199)	229.99	2.93	0.89	1.96	2.93	0.89	1.76	1.17	0.00	2.34	1.17	0.00	2.33
C10 (120)	180.09	1.64	1.65	0.16	1.64	1.65	0.50	1.64	1.65	0.01	1.64	1.65	0.00
C11 (100)	131.63	0.00	0.01	0.81	0.00	0.01	2.62	0.00	0.01	0.81	0.00	0.01	2.62
Medium													
G1 (240)	912.50	0.00	0.01	5.49	0.00	0.01	5.49	0.00	0.01	5.49	0.00	0.01	5.49
G2 (320)	1406.52	0.00	0.35	3.98	0.07	0.35	3.98	0.07	0.35	3.20	0.07	0.35	3.20
G3 (400)	1970.04	0.00	0.60	1.26	0.39	0.55	1.05	0.39	0.63	1.26	0.39	0.63	1.26
G4 (480)	2438.69	0.00	1.74	1.46	0.31	2.10	2.84	2.13	4.79	1.52	2.13	4.76	3.12
G5 (200)	1114.95	3.71	3.19	2.44	3.66	2.40	0.00	3.62	2.82	0.64	3.62	2.73	0.64
G6 (280)	1452.90	0.69	0.50	1.66	0.69	0.30	1.69	0.81	0.00	1.59	0.81	0.46	1.59
G7 (360)	1821.87	1.77	2.37	0.00	1.77	1.28	1.02	1.71	2.24	1.22	1.71	1.24	1.22
G8 (440)	2035.12	0.25	0.86	1.74	0.00	0.72	1.74	0.57	0.56	1.74	0.37	0.58	1.74
G9 (255)	98.91	1.42	0.00	0.42	1.42	0.65	0.06	1.42	0.95	0.29	1.42	0.02	0.06
G10 (323)	127.23	1.70	0.20	0.00	1.70	0.44	0.10	1.00	0.65	0.00	1.00	1.23	0.10
G11 (399)	158.56	1.82	0.52	0.16	1.82	0.52	0.01	1.82	0.52	0.16	1.82	0.52	0.00
G12 (483)	191.11	0.74	0.71	0.66	0.74	0.00	1.15	1.02	0.67	1.03	1.02	0.33	1.15
G13 (252)	142.51	0.00	0.41	0.82	0.00	0.93	0.53	0.00	0.41	0.79	0.00	0.93	0.43
G14 (320)	181.10	0.36	0.00	0.52	0.36	0.22	0.86	0.36	0.00	0.87	0.36	0.27	0.67
G15 (396)	224.56	0.29	0.17	0.85	0.29	0.39	0.84	0.29	0.00	0.85	0.29	0.30	0.82

Continued on next page

TABLE 6.2: – *Continued from previous page*

# (n)	Best Z	CR(C1)	LR(C1)	LC(C1)	CR(C2)	LR(C2)	LC(C2)	CR(M1)	LR(M1)	LC(M1)	CR(M2)	LR(M2)	LC(M2)
G16 (480)	270.61	1.30	0.08	1.31	1.30	0.30	1.25	1.30	0.00	1.31	1.30	0.74	1.28
G17 (240)	119.27	1.95	1.43	0.52	1.95	2.11	0.13	1.93	1.64	0.52	1.66	2.11	0.00
G18 (300)	168.60	0.07	0.68	0.86	0.00	0.61	0.06	0.00	0.68	0.75	0.00	0.61	0.15
G19 (360)	230.61	0.22	0.90	0.02	0.49	1.12	0.04	0.52	0.61	0.00	0.49	1.28	0.17
G20 (420)	308.77	0.48	0.40	1.45	0.43	0.00	1.35	0.20	0.72	1.45	0.15	0.47	1.35
						Large							
L1 (560)	2943.60	2.74	0.27	0.19	2.79	0.00	0.03	2.79	0.10	0.52	2.79	0.98	0.49
L2 (600)	2442.73	0.75	0.69	0.00	0.75	0.71	0.00	0.75	0.69	0.00	0.75	0.71	0.00
L3 (640)	3428.77	1.12	0.75	0.33	0.12	0.60	0.05	0.70	0.00	0.36	0.70	0.27	0.35
L4 (720)	3934.17	3.82	3.14	0.27	3.82	2.37	0.16	4.29	3.14	0.12	4.29	3.04	0.00
L5 (760)	2819.38	0.75	0.13	1.11	0.75	0.00	1.28	0.75	0.06	1.23	0.75	0.04	1.21
L6 (800)	4288.95	4.04	2.97	0.00	4.12	1.53	0.00	3.67	2.64	0.15	3.67	2.34	0.14
L7 (840)	2890.58	1.26	1.06	0.00	1.26	0.89	0.01	1.40	1.30	0.05	1.40	1.95	0.08
L8 (880)	4817.04	2.52	1.62	0.02	2.40	0.67	0.71	2.29	0.83	0.36	1.75	0.59	0.00
L9 (960)	5199.08	4.76	2.75	1.16	4.88	3.62	0.00	4.27	4.14	1.82	4.26	4.00	0.44
L10 (1040)	5802.05	2.20	0.71	0.05	1.71	0.37	0.01	0.14	0.05	0.04	0.21	0.66	0.00
L11 (1120)	6189.79	3.97	3.48	0.00	3.64	3.25	0.34	2.93	2.57	0.60	2.60	2.90	0.32
L12 (1200)	6756.20	1.29	1.15	0.91	1.08	0.16	0.34	0.30	1.42	0.92	0.22	1.44	0.00

Details of the computational results when $\alpha = 0.6$ are also exhibited in Table 6.2. It shows the percent deviation against the best Z for each instance in the small, medium and large data sets using twelve methods of the composite and multi-level heuristics. Other details can be found in Appendix B.

The summary result for the average percent deviation against the best Z from Table 6.1 is illustrated in Figure 6.6. It shows the average percent deviation of using six of each the composite and the multi-level heuristics in the small, medium and large data sets. The average of all instances is also recorded. It seems that the pattern in Figure 6.6(a) is roughly the same compared to Figure 6.6(b) although the percentage scale in (a) is a little bit higher. In other words, we can say that, on average, the solution using the multi-level heuristic is proportional to the solution from the composite heuristic given that the same mechanism (e.g. variant, initial solution, type of operator) is used. The ratio between them is reflected to be very small.

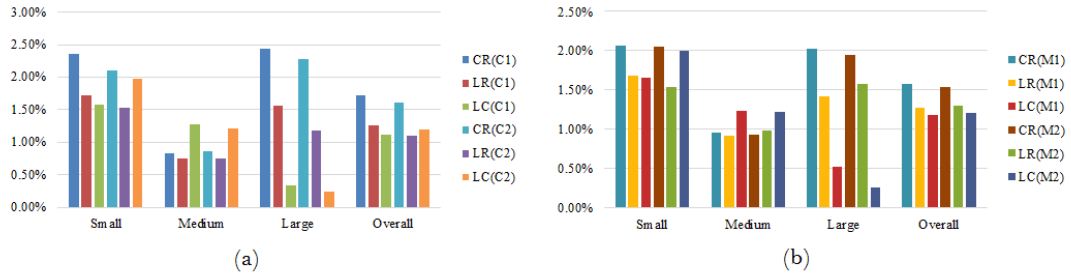

FIGURE 6.6: Average percent deviation against the best Z for the (a) composite and (b) multi-level heuristics for $\alpha = 0.6$

CPU times

In terms of the computational times, this is recorded at the start of the integrated search procedure until the final solution is developed. That means, once the savings method produced a solution, we applied the 2-opt operator and then record it as the initial solution. The start time is logged afterwards just before commencing the composite or the multi-level heuristics procedure. We displayed the average computational time for the composite and multi-level heuristics in Tables 6.3 and 6.4, respectively.

Overall, it can be observed that the average computational time of all instances is increasing when the heuristic uses CR, LR and LC. Furthermore, the speed of processing the heuristic of type 2 (i.e. C2 and M2) is higher than the one of type

TABLE 6.3: Average computational time for the composite heuristic (seconds)

Data set	CR(C1)	LR(C1)	LC(C1)	CR(C2)	LR(C2)	LC(C2)
Small	0.069	0.820	0.814	0.085	0.871	0.709
Medium	30.014	161.117	156.839	27.587	183.588	158.295
Large	2602.323	3279.026	3524.390	2572.292	3619.995	3768.731
Overall	691.937	925.699	987.845	683.053	1024.433	1052.187

1 (i.e. C1 and M1). This suggests that the search is more challenging when the 2-opt is applied within whenever there is an improvement of the solution in each step (or level). A similar observation to the one in the previous chapters, is also found for the large data set when the search spends a much longer time than in the case of the medium and the small data sets. This implies that the complexity of the search in the large data set is noticeably higher than the other two.

TABLE 6.4: Average computational time for the multi-level heuristic (seconds)

Data set	CR(M1)	LR(M1)	LC(M1)	CR(M2)	LR(M2)	LC(M2)
Small	0.068	0.847	0.868	0.072	0.784	0.926
Medium	17.500	149.407	143.448	18.915	165.899	165.691
Large	1889.771	2350.574	2628.766	1979.201	2452.274	3055.670
Overall	500.613	678.410	748.398	524.559	712.092	869.452

More importantly, on overall, we can observe that the multi-level heuristic is relatively faster than the composite one. This may be because of their differences in the refinement structure especially in the way the search is directed out of local optima.

6.3 The incorporation of learning mechanism in the integrated search

In this section, we present the learning as part of the search followed by some computational results.

6.3.1 Overview of the approach

Once a new better solution is found from the selected improvement scheme, it is not necessary to choose the next scheme in sequence or go back to the first level. It is possible that such solution is most apt to be with a certain improvement scheme. However, this can be a difficult task to pair up the solution with its matching scheme as we need to reflect on the solution construction and the improvement schemes attributes. The idea is to incorporate some degree of guidance in the search. In other words, a local search that shows to be promising ought to be used more than the one with little effect.

For that reason, we decided to integrate a learning stage to guide the search to be more adaptive and relatively more effective. Here, we set up a random choice to select the next local search based on a certain probability test.

How a local search is selected

This is based on the inverse method. Given $\upsilon \in [0,1]$ which is a randomly generated variable, we can choose the local search by $Loc_h^* = ArgF^{-1}(\upsilon)$ with F being the cumulative probability. Figure 6.7 illustrates the selection of a local search operator using the cumulative probability with respect to a randomly generated υ. For example, if $\upsilon = x$ then the selected local search is $Loc_h^* = Loc_3$. In our experiments, we have $h_{max} = 5$ and $(Loc_1, Loc_2, Loc_3, Loc_4, Loc_5) = ((1,0),(1,1),(2,0),(2,1),(2,2))$.

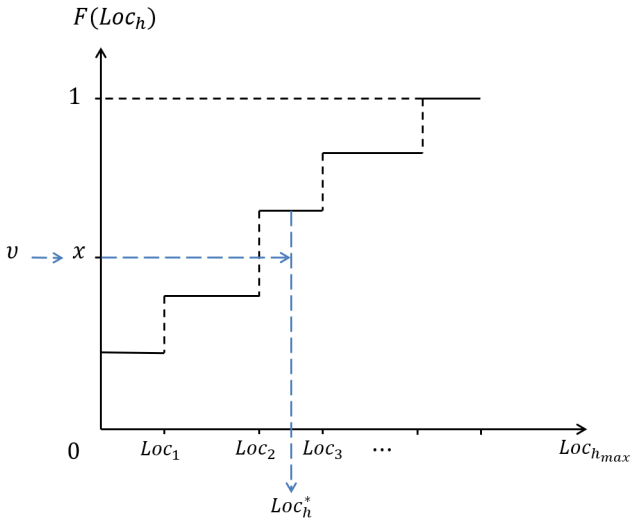

FIGURE 6.7: Selection of local search using the cumulative probability

Various test probability sets

We introduced three variants of the learning mechanism with respect to the different set of test probability. This is explained briefly as follows:

(a) **L1**: This learning method uses test probability generated by using a uniform distribution. For instance, in the proposed composite heuristic, we have five improvement schemes, therefore the probability ($p \in \{0, 1\}$) is divided uniformly between five of them. As for the multi-level method, the probability is spread out between four levels of improvement excluding the first level as the search is set to revert back to Level 1 if the solution is improved, due to the characteristics of this metaheuristic.

(b) **L2**: The second learning method calibrates the test probability based on the performance from a set of instances. This set of instances is called the training set. We selected 10 out of 46 instances in proportion from the small, medium and large data sets. We run the experiments using these instances and make note of which improvement schemes that provide an improvement to the solution. We referred to this as the score of improvement. We exhibited the score of improvement for LC(C2) and LC(M2) in Tables 6.5 and 6.6. From these outcomes, we acquired the test probability and the whole experiment was run again.

TABLE 6.5: Score of improvement for LC(C2)

# (n)	Score 1	Score 2	Score 3	Score 4	Score 5
F1 (44)	13	5	5	0	0
C6 (75)	8	5	1	0	0
C11 (100)	7	5	0	0	0
G2 (320)	10	8	3	2	2
G8 (440)	15	11	3	7	4
G12 (483)	14	13	4	5	5
G17 (240)	8	9	1	7	1
L3 (640)	17	11	11	7	7
L7 (840)	16	14	6	5	2
L11 (1120)	26	21	20	9	4
$\sum$	134	102	54	42	25
Prob.	0.38	0.29	0.15	0.12	0.07
Cum. Prob. (F)	0.38	0.66	0.81	0.93	1.00

(c) **L3**: The third learning method is implemented by adapting an adjusted probability within the search. This method is composed of two phases; the first phase is conducted using the probability as in (a) for $k \cdot m$ times with k being the number of improvement schemes used in the procedure and m the number of times the scheme is repeated. In our experiments, we set $k = 5$ and $m = 1$. The score of improvement is recorded by setting $score = score + 1$ and the test probability is adjusted accordingly. Next, in the second phase, the procedure continues to select the improvement scheme based on the updated probability. While doing that, the score of improvement is also recorded. The search goes on until it reaches another $k \cdot m$. Then the test probability is updated once again with the latest score of improvement. The search continues with the renewed test probability and this method stops when no further improvement can be found after all schemes have been applied.

Table 6.6: Score of improvement for LC(M2)

# (n)	Score 2	Score 3	Score 4	Score 5
F1 (44)	0.33	0.50	0.00	0.00
C6 (75)	0.50	0.00	0.00	0.00
C11 (100)	0.50	0.00	0.00	0.00
G2 (320)	0.57	0.00	0.22	0.29
G8 (440)	0.40	0.17	0.40	0.44
G12 (483)	0.61	0.00	0.22	0.29
G17 (240)	0.46	0.13	0.54	0.17
L3 (640)	0.43	0.54	0.25	0.44
L7 (840)	0.60	0.00	0.38	0.00
L11 (1120)	0.53	0.55	0.33	0.17
$\sum$	4.93	1.89	2.34	1.80
Prob.	0.45	0.17	0.21	0.16
Cum. Prob. (F)	0.45	0.62	0.84	1.00

6.3.2 Analysis and computational result

The computational experiments to test the proposed learning mechanism are conducted using the composite and multi-level heuristics of the variants LC(C2) and LC(M2). These variants are selected because they show promising resuts in the previous section. This will support the 'guided' search to produce a better quality

solution. Three types of learning schemes (i.e. L1, L2 and L3) are incorporated within the search leading to eight methods in total including the LC(C2) and LC(M2).

The computational tests were accomplished using the same 46 instances of the well-known VRP test problems which are categorised in the small, medium and large data sets. In this test, the value of α is set at 0.6 and the search using each method is run for five times. Statistically, five runs of the test are considered as small and would produce limited outcomes, but due to the time constraint, we thought it is sufficient to get interesting information based on this smaller search.

To evaluate the performance of the learning mechanism, we assessed the solution through the measure of the percent deviation compared to the best solution obtained from among the methods. The lower the percent deviation, the more attractive the solution becomes. This reflects the solution being nearer to the best one. For a given instance, the percent deviation of an average and the best (i.e. the minimum value) solution are calculated with respect to the five experimental runs of each method. Table 6.7 provides the detailed computational results for each of the eight methods and for each instance.

Subsequently, the results are summarised as presented in Table 6.8. The data shown the average percent deviation for the small, medium, and large data sets as well as for the overall instances. The frequency of the instances using each method that produced the best Z is recorded as the total best Z. It is worth observing the lowest average percent deviation as this implies that there are more instances in the data set with the best Z or the solution near it. For the small data set, the best solution of LC(C2L3) recorded the lowest average percent deviation which is at 0.20%. The same method also produced the lowest overall average percent deviation (0.44%). In addition, it is found to have the largest count (which is 14) of the solution with the best Z. For the medium and the large data sets, the average percent deviation is found to be the lowest when the methods LC(M2L2) and LC(C2L2) are implemented, respectively.

TABLE 6.7: Detailed percent deviation (in %) against the best Z of the learning mechanism using the composite and multi-level heuristics for $\alpha = 0.6$

# (n)	Best Z	LC(C2)	LC(C2L1)		LC(C2L2)		LC(C2L3)		LC(M2)	LC(M2L1)		LC(M2L2)		LC(M2L3)	
			Avg	Best	Avg	Best	Avg	Best		Avg	Best	Avg	Best	Avg	Best
Small															
F1 (44)	113.28	6.37	4.14	0.90	4.20	0.94	2.47	0.00	6.37	4.17	0.87	2.98	0.85	4.17	0.87
F2 (71)	41.59	0.07	1.04	0.05	1.03	0.00	1.11	0.07	0.07	0.63	0.00	0.63	0.05	0.63	0.05
F3 (134)	186.84	2.89	2.21	1.70	2.34	1.80	1.33	0.00	2.89	2.63	2.17	2.45	1.69	2.53	1.86
C1 (21)	58.20	0.05	3.41	0.02	4.97	0.05	2.59	0.00	0.05	3.26	0.00	3.26	0.00	3.26	0.00
C2 (22)	95.20	6.75	5.38	0.00	5.48	0.09	5.27	0.08	6.75	7.67	6.75	7.67	6.75	7.67	6.75
C3 (29)	74.85	4.93	4.52	1.31	4.53	1.31	3.61	0.00	4.93	4.53	1.31	4.53	1.31	4.53	1.31
C4 (32)	178.47	0.39	0.75	0.01	1.05	0.03	0.74	0.00	0.39	1.05	0.03	1.05	0.03	1.05	0.03
C5 (50)	86.82	3.05	4.25	3.05	3.75	0.00	2.38	0.09	3.05	4.68	3.05	4.68	3.05	4.68	3.05
C6 (75)	134.84	2.48	3.76	3.34	3.60	3.34	3.56	0.00	2.48	3.07	2.48	3.38	2.48	3.07	2.48
C7 (100)	138.03	8.39	2.87	0.43	3.84	2.38	2.98	0.00	8.39	4.31	2.99	4.01	1.61	4.59	3.47
C8 (150)	168.77	2.50	6.05	1.68	5.14	3.48	3.42	0.00	2.72	3.48	2.83	4.97	3.31	5.34	2.86
C9 (199)	224.52	4.24	3.63	0.00	2.74	1.74	1.98	0.51	4.83	4.57	3.17	4.66	2.90	4.66	2.51
C10 (120)	165.15	9.59	9.99	8.03	8.50	3.54	6.06	1.80	9.05	4.73	0.03	2.74	0.00	3.08	0.03
C11 (100)	132.33	2.08	1.42	0.68	1.51	0.26	0.94	0.17	2.08	1.16	0.00	1.16	0.00	1.16	0.00
Medium															
G1 (240)	962.42	0.02	0.22	0.19	0.20	0.00	0.23	0.19	0.02	0.21	0.02	0.21	0.02	0.21	0.02
G2 (320)	1451.04	0.79	1.30	0.84	1.30	1.07	1.25	0.71	0.04	1.24	0.63	0.96	0.00	0.87	0.00

Continued on next page

TABLE 6.7: – *Continued from previous page*

# (n)	Best Z	LC(C2)	LC(C2L1)		LC(C2L2)		LC(C2L3)		LC(M2)	LC(M2L1)		LC(M2L2)		LC(M2L3)	
			Avg	Best	Avg	Best	Avg	Best		Avg	Best	Avg	Best	Avg	Best
G3 (400)	1984.46	0.32	0.54	0.19	0.46	0.39	0.51	0.38	0.52	0.38	0.26	0.36	0.00	0.41	0.31
G4 (480)	2436.31	2.94	2.10	0.60	3.76	2.53	2.15	0.73	3.22	1.08	0.00	1.71	0.41	2.03	0.23
G5 (200)	1109.78	0.47	3.04	0.05	4.09	3.07	3.56	2.83	1.11	2.77	0.51	3.09	0.00	3.42	0.51
G6 (280)	1476.03	0.10	2.00	0.70	1.75	0.75	2.25	1.64	0.00	1.60	1.14	1.36	0.70	1.64	1.14
G7 (360)	1827.77	0.70	0.52	0.40	0.72	0.05	0.65	0.44	0.89	0.57	0.00	0.61	0.27	0.61	0.14
G8 (440)	2026.77	2.16	1.71	1.50	1.51	0.82	1.87	1.42	2.16	1.69	0.00	1.52	0.00	1.64	0.00
G9 (255)	98.79	0.18	0.72	0.25	1.92	0.16	1.59	0.00	0.18	1.17	0.18	1.11	0.18	1.03	0.18
G10 (323)	125.13	1.78	2.25	0.09	1.70	1.04	2.39	0.00	1.78	1.65	0.97	2.08	0.68	2.17	1.51
G11 (399)	157.28	0.82	1.58	0.72	1.55	-0.15	1.58	0.06	0.81	1.13	0.00	1.83	1.25	1.90	0.00
G12 (483)	191.74	0.81	1.12	0.00	1.32	0.28	1.18	0.00	0.81	1.34	0.81	1.85	0.81	1.30	0.58
G13 (252)	142.54	0.51	1.03	0.00	0.98	0.41	1.08	0.23	0.41	1.47	0.41	1.10	0.41	1.24	0.41
G14 (320)	181.33	0.73	1.32	0.47	1.42	0.00	1.27	0.60	0.54	1.51	0.44	1.54	0.44	1.36	0.44
G15 (396)	225.53	0.40	1.08	0.11	1.28	0.19	1.02	0.00	0.39	1.11	0.29	0.74	0.12	1.32	0.29
G16 (480)	272.99	0.36	1.52	0.15	1.42	0.64	1.40	0.11	0.40	2.10	0.23	1.52	0.00	1.66	0.29
G17 (240)	117.66	1.50	3.26	2.69	3.01	1.77	2.23	0.42	1.37	1.78	0.89	1.18	0.00	1.82	0.89
G18 (300)	167.69	0.60	0.33	0.00	0.63	0.28	0.43	0.00	0.70	0.44	0.23	0.78	0.48	0.41	0.14
G19 (360)	228.60	0.92	1.18	0.72	1.30	0.79	1.08	0.72	1.05	1.16	0.00	1.00	0.87	1.22	0.77
G20 (420)	310.58	0.76	0.42	0.00	0.54	0.13	0.59	0.20	0.76	0.51	0.09	0.43	0.14	0.38	0.17

Continued on next page

TABLE 6.7: – *Continued from previous page*

# (n)	Best Z	LC(C2)	LC(C2L1)		LC(C2L2)		LC(C2L3)		LC(M2)	LC(M2L1)		LC(M2L2)		LC(M2L3)	
			Avg	Best	Avg	Best	Avg	Best		Avg	Best	Avg	Best	Avg	Best
							Large								
L1 (560)	2890.65	1.87	2.02	1.09	1.59	0.00	1.71	1.09	2.33	2.17	1.48	1.99	1.50	2.21	1.95
L2 (600)	2431.94	0.44	0.66	0.00	0.60	0.06	0.70	0.00	0.44	0.79	0.58	0.80	0.59	0.79	0.58
L3 (640)	3402.99	0.81	1.13	0.57	1.09	0.74	1.12	0.72	1.11	0.88	0.10	0.56	0.00	0.75	0.18
L4 (720)	3922.84	0.45	1.21	0.53	1.22	0.00	1.27	0.19	0.29	1.67	0.31	1.69	0.36	1.62	0.31
L5 (760)	2841.06	0.50	0.81	0.59	0.68	0.00	0.91	0.59	0.44	0.72	0.23	0.65	0.23	0.71	0.23
L6 (800)	4276.60	0.29	1.23	0.35	0.93	0.00	0.79	0.28	0.43	0.35	0.30	0.37	0.29	0.35	0.30
L7 (840)	2870.96	0.69	0.87	0.00	0.97	0.45	0.87	0.45	0.76	1.14	0.70	1.20	0.70	1.12	0.70
L8 (880)	4799.90	1.07	2.64	2.08	1.47	0.64	1.70	1.21	0.36	0.53	-0.11	0.59	0.00	0.78	0.19
L9 (960)	5199.08	0.00	1.31	0.80	1.09	0.53	1.34	0.53	0.44	1.74	0.53	1.48	0.51	1.43	0.53
L10 (1040)	5759.24	0.75	1.14	0.71	0.95	0.00	0.90	0.02	0.74	0.77	0.68	0.70	0.64	0.73	0.49
L11 (1120)	6161.86	0.80	1.45	0.52	1.62	0.00	1.56	0.55	0.78	1.79	1.10	1.67	1.01	1.73	1.10
L12 (1200)	6756.20	0.34	2.09	1.34	1.87	0.62	2.15	1.35	0.00	2.07	0.61	1.95	0.05	2.03	0.48

TABLE 6.8: Summary table for the average percent deviation (in %) and the total best Z of the learning mechanism using the composite and multi-level heuristics

Data set	LC(C2)	LC(C2L1)		LC(C2L2)		LC(C2L3)		LC(M2)	LC(M2L1)		LC(M2L2)		LC(M2L3)	
		Avg	Best	Avg	Best	Avg	Best		Avg	Best	Avg	Best	Avg	Best
Small	3.84	3.82	1.51	3.76	1.36	2.75	**0.20**	3.86	3.57	1.84	3.44	1.72	3.60	1.81
Medium	0.84	1.36	0.48	1.54	0.71	1.42	0.53	0.86	1.24	0.36	1.25	**0.34**	1.33	0.40
Large	0.67	1.38	0.71	1.17	**0.25**	1.25	0.58	0.68	1.22	0.54	1.14	0.49	1.19	0.59
Overall	1.71	2.11	0.86	2.12	0.79	1.78	**0.44**	1.72	1.94	0.85	1.89	0.80	1.99	0.88
Total Best Z	1	0	8	0	10	0	**14**	2	0	8	0	11	0	4

Further illustration can be viewed in Figure 6.8. It can be observed that the average solution among the five test runs always produced a higher solution than the best one, in terms of the percent deviation compared to the best Z found in among the methods. This may be because the test probability of choosing the next refinement module is selected randomly. Therefore, among the test runs, some of them have a good solution and some have not.

CPU times

The computing time for the learning mechanism is recorded when the procedure of the integrated search starts. The end time is logged when the final solution is formed. The computing time is also recorded over five test runs. The result is demonstrated in Table 6.9. It can be observed that the overall average time for an instance to complete the search is very long, with the lowest one is about 1.5 hour (5270.697 seconds). Furthermore, the learning mechanisms using the multi-level heuristic (M2) is found to be faster than using the composite heuristic (C2).

TABLE 6.9: Average computational time for learning mechanism over five runs (seconds)

Data set	LC(C2L1)	LC(C2L2)	LC(C2L3)	LC(M2L1)	LC(M2L2)	LC(M2L3)
Small	7.203	5.734	6.562	3.652	3.669	**3.455**
Medium	323.398	445.392	379.510	381.420	**348.556**	350.634
Large	29857.556	23123.971	36871.987	21229.485	**19619.133**	20231.396
Overall	7931.727	6227.734	9785.780	5705.073	**5270.697**	5431.257

6.4 Summary

In this chapter, we investigated the performance of the integrated search consisting of the combination of the local search operators using the composite and multi-level heuristics. Two variants of the heuristics are performed in including the

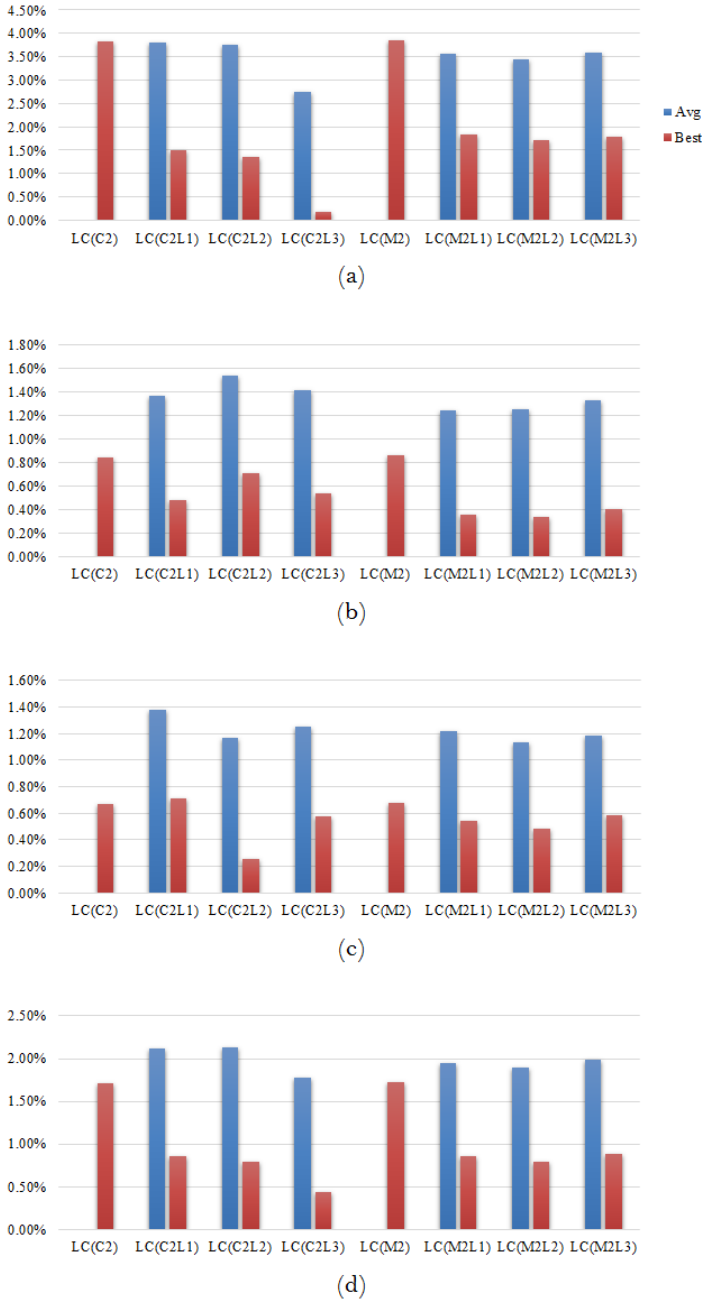

FIGURE 6.8: Average percent deviation against the best Z for the learning mechanism using the composite and multi-level heuristics for (a) small, (b) medium, and (c) large data sets, as well as the (d) overall instances

2-opt operator within the search. Extensive computational tests were undertaken using the 46 well-known VRP instances. Interesting results of both heuristics are obtained. This test inspired us to incorporate the learning mechanism in the search to encourage the solution to be relatively more efficient. We highlighted

three types of the learning mechanisms with respect to the test probability in choosing the successive improvement scheme. These result in some interesting findings. In the next chapter, we draw the conclusion of this study and provide some suggestions to improve and expand this research.

Chapter 7

Conclusions and Suggestions

7.1 Introduction

This chapter provides an overall summary of the research and accentuates some research prospects that might be worth exploring in the future.

7.2 Conclusions

This study examines the impact of the load factor in heuristic search for the case of the environmental routing problem. In other words, the classical vehicle routing problem (VRP) namely the capacitated VRP (CVRP) with the incorporation of the load factor is analysed. Three types of heuristic searches are explored which include the constructive heuristic, the local search and the integrated heuristic. New formulations for the selected constructive method and the local searches with the attachment of the load factor are introduced. In the integrated heuristic, a set of the local search operators is utilised altogether as it gives a better alteration to the solution rather than using a single operator. A learning mechanism is then introduced to guide the search to be comparatively more efficient. The effectiveness and flexibility of the new load-based formulations are shown to be a useful addition to the local searches as demonstrated by the interesting solutions obtained when tested on the 46 well-known VRP data sets.

This thesis is organised into seven chapters. Each chapter is briefly described in the following. Starting with the first chapter, here we provided an overview of the routing problems focusing on the transportation planning. As the recent

transportation activities rise to a greater concern of the environmental impact, we highlighted issues associated with CO_2 emissions and its relationship with the load factor. In addition, we gave a brief description of the common research methodology to solve the routing problems.

In Chapter 2, we investigated some of the literature related to this research. This includes an overview of VRP especially the capacitated one and its basic optimisation model. Some review of researches from the area of VRP are also embraced, especially the ones associated with sustainable transportation activities. This corresponds to another class of VRP named the green VRP, which includes pollution routing problem and VRP with reverse logistic. We also explored some related literature to the estimation of vehicle fuel consumption and emission. This estimation method is essential in this research as we adopted it to approximate the routing cost in the subsequent chapters.

Thereafter in Chapter 3, one of the most studied constructive heuristics namely, Clarke and Wright savings is analysed. We introduced two new savings formulations to incorporate the load factor known as the integrated and the combined savings. The emission model known as the Comprehensive Modal Emission Model (CMEM) is adapted to be the cost function to calculate the fuel consumption in measuring the CO_2 emission. Some adjustments to the model are also discussed. We also highlighted the evaluation of the reverse route to put more weight on the load factor that is being investigated. This scheme remained to be of the essence throughout this research. Computational experiments performed using the 46 of the well-known VRP instances generated stimulating results. It is found that the combined savings performed better than the integrated one. Therefore, the solution obtained from the combined savings along with the non-load-based regular savings is used as the initial solution in the rest of the thesis.

Subsequently, Chapter 4 studies the first three basic local search operators that are the 2-opt, (1,0) relocate and (2,0) relocate. The computation of these operators is generally simple and relatively quick. The 2-opt is performed only for the intra-route case, but all other operators are performed for both the intra-route and the inter-route cases. We presented new mathematical formulations which take into account the load factor for the first time. An illustrative example is then provided followed by full computational results obtained using the same VRP instances. In addition, we also highlighted the computational comparison between the brute-force and the formula-based searches. It is found that a large gap between the

two in terms of computing time exists. This finding encourages us to develop new formulations with the attached load factor for the search process.

In Chapter 5, three more intensive local search operators namely, the (1,1), (2,1) and (2,2) interchange are analysed. These operators are considered as advanced as they require relatively much more computing effort. Both the intra-route and the inter-route cases are implemented separately for a single operator. Similar to what was done in Chapter 4, the load factor is adapted to obtain new mathematical formulations. An illustrative example to illustrate the operator is also given. This is followed by an extensive computational experiment conducted using the renowned 46 VRP instances. Interesting outcomes are observed. Among the operators, there is no clear supremacy with respect to their solution quality.

Chapter 6 analyses the performance when all these local search operators are put altogether in the search. The method, which is referred to as the integrated search, comprised of two powerful mechanisms namely; the composite and the multi-level heuristics. These two metaheuristics refine the solution by using improvement schemes, also known as the local searches operators. These are arranged in a certain order which can be critical. For simplicity, we ranked them according to their time complexity leading to the following order: (1,0), (1,1), (2,0), (2,1), and (2,2). These two methods are quite similar to each other, apart from the way it directed the search once a particular improvement scheme is implemented. In this study, we provided a brief description of the two methods as well as the algorithms which led to twelve variants. This is followed by the computational results acquired with the use of 46 well-known VRP instances over all the variants. Two of these variants are found to be prominent and they are selected for the next stage where the learning mechanism is then introduced. We established a random choice based on a certain test probability to select the next local search, and we came out with three sets of test probability to be incorporated in the integrated search. Computational tests based on the 46 known data sets are performed over five runs with interesting outcomes. It was found that in general, for small and large data sets, the learning search within the composite method yielded a better solution. However, for the medium data set, the learning search within the multi-level heuristic performed better. It was also noted that in terms of computational time, the learning search within the multi-level heuristics is mostly faster than the composite one.

7.3 Future research directions

In this section, we point out some research avenues that we think will be worthwhile to pursue in the future. We would also like to emphasise that some of these points have already been briefly expressed in their respective chapters. These avenues are classified under two categories; the first being based on the possible enhancements that can be made on the proposed methods, whereas the second is focused on the general problem.

7.3.1 Improvement of the proposed method

In this study, we concentrate on the classical VRP which restricts the search to its vehicle capacity. However, the effect of the load factor could also be applied to other routing problems with additional characteristics and constraints. For example, the CVRP can be expanded to perform both tasks of delivery and collection simultaneously with the incorporation of the load factor in its algorithm.

Another issue that would be useful to explore is the adaptation of other constructive heuristics in obtaining the initial solution while introduces the load factor. For instance, the formulation of the nearest neighbourhood search could be easily modified to include the load factor with the measure of fuel consumption function. By manipulating several methods to obtain the initial solution, this can create a wide selection of neighbourhoods to be used for further improvement.

While using the same model to estimate the fuel consumption, it is also worthwhile to explore the effect of other factors such as the vehicle speed, acceleration and road gradient to gain more accuracy on the calculation. For example, instead of defining the speed as the average along the network, one could create a predefined speed matrix that may reflect on different types of the road travelled by the vehicle. Additionally, a positive road gradient results in an increase in fuel consumption and it should be carefully considered in impending route planning. The latest GIS software is able to cater for the road gradient data.

Moreover, it may be worthy to employ another model to estimate the fuel consumption or the CO_2 emission to calculate the cost of the route. From the literature, there are quite a few more emission models that can be explored such as

the Computer programme to calculate emissions from road transportation (COPERT). It can be noted that the success of employing such model might depend on the availability of data.

Another aspect that might be worth delving into is the application of other local search operators. The load factor or even other factors that affecting the fuel consumption measure could be integrated into their formulation with the same principle as what we have done here. The λ-opt, μ-relocate, and μ-ψ interchange operators can be benefited by using a different value of λ, μ and ψ. To begin with, we would likely to suggest applying $\lambda = 3$ as it is found to be suitable for many applications in the literature. A more complex local search procedure might also be worthwhile so long as there is a close monitor to ensure the search remains computationally acceptable.

Other than that, it might be interesting to have a powerful metaheuristic implemented in the search process. One possible method is to apply variable neighbourhood search (VNS) which is not far from the multi-level method. A genetic algorithm could also be attempted as we can expand the local searches to learn the operation of this method by applying the crossover, mutation and reproduction procedure. Extra care needs to be taken as these population type methods can consume more time.

Moreover, in this study, we applied the 2-opt operator within the integrated search whenever an improvement is found. It would be useful and possibly can reduce a little bit of time if the 2-opt is applied only to the affected route(s). Note that, it can be challenging as it is difficult to identify such route(s) as one route may experience many times of change in one procedure.

Furthermore, in the learning stage, we conducted a small test runs but this can easily be expanded to have a larger experiment with say, 30 reps to have better statistical results. Also, other α values could also be experimented to see the effect of the weight applied to the distance and fuel consumption costs.

In this study, we made adjustments to the well-known VRP instances in terms of the vehicle capacity and customers' demand to have a meaningful effect when using the selected fuel consumption function. Thus, it would be interesting to attempt this kind of problem in a real case study. Companies and business organisations can benefit from the findings to provide more efficient transportation services to their clients.

We would also like to emphasise that in our research, the vehicle load impact very much depends on the distance measure. The effect of the load can then be distorted or misleading as the distance also influences the decision. Having a measurement which specifically caters for the load function and which is not restricted to the distance only, is a worthwhile task to pursue as its effect of the load may be practically more significant.

7.3.2 General improvement

Further exploration could be attempted by transforming the problem into a multi-objective problem with more than two objective functions. More objectives with regards to different criteria could be added so as to have a comprehensive solution closer to the nature of the problem. This problem then can be tackled with multi-objective optimisation techniques such as the evolutionary algorithm to approach the Pareto frontier.

For a given method, it is also important to understand each and every step in the procedure as this will allow us to design a new data structure that can minimise and avoid redundant calculations. This can lead to a saving in computational time without affecting the solution quality. Another time-saving approach that can be endeavoured is by constructing some neighbourhood reduction schemes as these help to avoid performing unnecessary calculations. These schemes would even be devised in a dynamic way with the aim to prioritise each type of neighbourhood and each type of local search. This deep learning is a very contemporary topic that is worth examining.

Appendix A

Contributions to knowledge

In this appendix, the main contributions that stem from this research are highlighted. These include conference items and potential academic papers.

Conference Items

1. Ahmad, N., Salhi, S., Acquaye, A. (2017) Adapting Constructive Heuristics and Local Search Operators in Green Vehicle Routing Problem. In *OR59 Annual Conference, 12-14 September 2017*. United Kingdom: Loughborough. [Part of chapters 3 and 4]

2. Ahmad, N., Salhi, S., Acquaye, A. (2018) Introduction of Load Factor in Local Search Operators. In *OR60 Annual Conference, 11-13 September 2018*. United Kingdom: Lancaster. [Part of chapters 4 and 5]

Academic Papers (in preparation)

1. Incorporation of the load factor in constructive VRP heuristics [Part of chapters 2 and 3]

2. The introduction of the load factor in local searches for combinatorial optimisation [Part of chapters 4, 5 and 6]

Appendix B

Detailed computational results of the composite and multi-level heuristics

In this appendix, a detailed computational results of the composite and multi-level heuristics are presented. It is shown in Table B.1, the objective function solution Z obtained with different values $\alpha = 0.6, 0.7, 0.8, 0.9, 1.0$.

TABLE B.1: Detailed computational results for the composite and multi-level heuristics

# (n)	α	C1			C2			M1			M2		
		CR	LR	LC	CR	LR	LC	CR	LR	LC	CR	LR	LC
							Small						
F1 (44)	0.6	116	112.67	120.08	112.14	114.42	120.5	112.14	112.67	120.08	112.14	114.42	120.5
	0.7	118.82	115.64	123.2	115.03	117.64	123.66	115.03	115.64	123.2	115.03	117.64	123.66
	0.8	121.63	118.62	120.76	117.93	120.86	120.86	117.93	118.62	120.76	117.93	120.86	120.86
	0.9	124.45	121.59	121.59	120.82	124.07	121.71	120.82	121.59	121.59	120.82	124.07	121.71
	1	127.27	124.56	124.56	123.72	127.29	127.29	123.72	124.56	124.56	123.72	127.29	127.29
F2 (71)	0.6	43.84	42.74	41.73	43.84	42.74	41.62	43.84	42.74	41.73	43.84	42.74	41.62
	0.7	44.91	43.79	42.79	44.91	43.79	42.64	44.91	43.79	42.79	44.91	43.79	42.64
	0.8	45.97	44.84	43.84	45.97	44.84	43.65	45.97	44.84	43.84	45.97	44.84	43.65
	0.9	47.03	45.89	44.9	47.03	45.89	44.66	47.03	45.89	44.9	47.03	45.89	44.66
	1	48.09	46.93	46.93	48.09	46.93	46.93	48.09	46.93	46.93	48.09	46.93	46.93
F3 (134)	0.6	191.36	190.96	192.24	191.07	191.35	192.24	191.36	190.96	192.24	191.07	191.35	192.24
	0.7	195.58	195.21	196.6	195.28	195.62	196.19	195.58	195.21	196.6	195.28	195.62	196.19
	0.8	199.8	199.46	199.92	199.49	199.89	199.53	199.8	199.46	199.84	199.49	199.89	199.09
	0.9	204.02	203.71	201.52	203.69	204.16	205.01	204.02	203.71	199.79	203.69	204.16	204.95
	1	208.24	207.97	207.97	207.9	208.43	208.43	208.24	207.97	207.97	207.9	208.43	208.43
C1 (21)	0.6	58.23	59.8	59.8	58.23	58.23	58.23	58.23	59.8	59.8	58.23	58.23	58.23
	0.7	59.62	61.17	61.17	59.62	59.62	59.62	59.62	61.17	61.17	59.62	59.62	59.62
	0.8	61.02	62.54	62.54	61.02	61.02	61.02	61.02	62.54	62.54	61.02	61.02	61.02
	0.9	62.41	63.92	63.92	62.41	62.41	62.41	62.41	63.92	63.92	62.41	62.41	62.41

Continued on next page

Table B.1: – *Continued from previous page*

# (n)	α	C1			C2			M1			M2		
		CR	LR	LC	CR	LR	LC	CR	LR	LC	CR	LR	LC
	1	63.8	65.29	65.29	63.8	63.8	63.8	63.8	65.29	65.29	63.8	63.8	63.8
C2 (22)	0.6	101.23	99.71	102.89	101.23	99.71	101.63	101.23	99.71	102.89	101.23	99.71	101.63
	0.7	104.18	102.69	105.94	104.18	102.69	104.62	104.18	102.69	105.94	104.18	102.69	104.62
	0.8	107.14	105.66	108.99	107.14	105.66	107.62	107.14	105.66	108.99	107.14	105.66	107.62
	0.9	110.1	108.64	112.03	110.1	108.64	110.61	110.1	108.64	112.03	110.1	108.64	110.61
	1	113.05	111.61	111.61	113.05	111.61	111.61	113.05	111.61	111.61	113.05	111.61	111.61
C3 (29)	0.6	78.8	78.54	78.54	78.8	78.54	78.54	78.8	78.54	78.54	78.8	78.54	78.54
	0.7	80.69	80.49	80.49	80.69	80.49	80.49	80.69	80.49	80.49	80.69	80.49	80.49
	0.8	82.59	82.44	82.44	82.59	82.44	82.44	82.59	82.44	82.44	82.59	82.44	82.44
	0.9	84.49	84.39	84.39	84.49	84.39	84.39	84.49	84.39	84.39	84.49	84.39	84.39
	1	86.38	86.34	86.34	86.38	86.34	86.34	86.38	86.34	86.34	86.38	86.34	86.34
C4 (32)	0.6	179.35	180.62	179.16	179.35	180.62	179.16	179.35	180.62	179.16	179.35	180.62	179.16
	0.7	183.99	185.32	183.96	183.99	185.32	183.96	183.99	185.32	183.96	183.99	185.32	183.96
	0.8	188.64	190.03	188.76	188.64	190.03	188.76	188.64	190.03	188.76	188.64	190.03	188.76
	0.9	193.28	194.73	193.56	193.28	194.73	193.56	193.28	194.73	193.56	193.28	194.73	193.56
	1	197.93	199.43	199.43	197.93	199.43	199.43	197.93	199.43	199.43	197.93	199.43	199.43
C5 (50)	0.6	92.12	88.82	87.25	92.12	87.06	89.47	91.8	88.79	87.25	91.8	87.86	89.47
	0.7	94.13	90.75	89.22	94.13	88.96	91.47	93.87	90.73	89.22	93.87	89.81	91.47
	0.8	96.15	92.67	92.67	96.15	90.86	90.86	95.93	92.66	92.66	95.93	91.75	91.75
	0.9	98.16	94.6	94.6	98.16	92.76	92.76	98	94.6	94.6	98	93.69	93.69

Continued on next page

TABLE B.1: – *Continued from previous page*

# (n)	α	C1			C2			M1			M2		
		CR	LR	LC	CR	LR	LC	CR	LR	LC	CR	LR	LC
	1	100.18	96.53	96.53	100.18	94.66	94.66	100.07	96.54	96.54	100.07	95.63	95.63
C6 (75)	0.6	136.46	135.82	138.19	136.46	136.66	138.19	136.46	136.66	138.19	136.46	136.66	138.19
	0.7	139.39	138.74	141.24	139.39	139.64	141.24	139.39	139.64	141.24	139.39	139.64	141.24
	0.8	142.33	141.67	144.29	142.33	142.61	144.29	142.33	142.61	144.29	142.33	142.61	144.29
	0.9	145.27	144.6	151.36	145.27	145.59	151.36	145.27	145.59	151.36	145.27	145.59	151.36
	1	148.21	147.53	147.53	148.21	148.56	148.56	148.21	148.56	148.56	148.21	148.56	148.56
C7 (100)	0.6	143.96	147.42	138.27	143.96	147.42	149.61	147.24	147.42	143.41	147.24	147.42	149.61
	0.7	147.19	150.71	149.92	147.19	150.71	148.23	150.55	150.71	149.92	150.55	150.71	145.25
	0.8	150.43	154	153.22	150.43	154	151.54	153.86	154	153.22	153.86	154	148.48
	0.9	153.67	157.29	156.53	153.67	157.29	154.85	157.17	157.29	156.53	157.17	157.29	151.7
	1	156.91	160.58	160.58	156.91	160.58	160.58	160.48	160.58	160.58	160.48	160.58	160.58
C8 (150)	0.6	185.25	183.76	178.64	185.25	183.17	172.99	183.71	183.26	173.64	183.71	183.12	173.36
	0.7	189.38	187.92	182.72	189.38	187.27	176.93	187.79	187.36	177.59	187.79	187.18	177.31
	0.8	193.51	192.08	192.08	193.51	191.38	191.38	191.86	191.46	191.46	191.86	191.25	191.25
	0.9	197.65	196.24	196.24	197.65	195.49	195.49	195.94	195.56	195.56	195.94	195.32	195.32
	1	201.78	200.4	200.4	201.78	199.6	199.6	200.02	199.66	199.66	200.02	199.39	199.39
C9 (199)	0.6	236.72	232.03	234.5	236.72	232.03	234.04	232.67	229.99	235.37	232.67	229.99	235.36
	0.7	241.88	237.14	225.57	241.88	237.14	227.76	237.79	235.08	228.2	237.79	235.08	227.76
	0.8	247.04	242.26	251.5	247.04	242.26	251.66	242.91	240.16	251.5	242.91	240.16	251.66
	0.9	252.2	247.37	238.06	252.2	247.37	239.64	248.04	245.25	242.96	248.04	245.25	239.02

Continued on next page

Table B.1: – *Continued from previous page*

# (n)	α	C1			C2			M1			M2		
		CR	LR	LC	CR	LR	LC	CR	LR	LC	CR	LR	LC
	1	257.36	252.49	252.49	257.36	252.49	252.49	253.16	250.33	250.33	253.16	250.33	250.33
C10 (120)	0.6	183.04	183.07	180.37	183.04	183.07	180.09	183.04	183.07	180.11	183.04	183.07	180.09
	0.7	187.11	187.17	176.02	187.11	187.17	176.02	187.11	187.17	176.02	187.11	187.17	176.02
	0.8	191.19	191.27	179.81	191.19	191.27	179.81	191.19	191.27	179.81	191.19	191.27	179.81
	0.9	195.27	195.37	195.37	195.27	195.37	195.37	195.27	195.37	195.37	195.27	195.37	195.37
	1	199.35	199.47	199.47	199.35	199.47	199.47	199.35	199.47	199.47	199.35	199.47	199.47
C11 (100)	0.6	131.63	131.64	132.69	131.63	131.64	135.08	131.63	131.64	132.69	131.63	131.64	135.08
	0.7	134.6	134.62	135.62	134.6	134.62	138.09	134.6	134.62	135.62	134.6	134.62	138.09
	0.8	137.57	137.6	137.6	137.57	137.6	137.6	137.57	137.6	137.6	137.57	137.6	137.6
	0.9	140.53	140.58	140.58	140.53	140.58	140.58	140.53	140.58	140.58	140.53	140.58	140.58
	1	143.5	143.57	143.57	143.5	143.57	143.57	143.5	143.57	143.57	143.5	143.57	143.57
Medium													
G1 (240)	0.6	912.5	912.59	962.59	912.5	912.59	962.59	912.5	912.59	962.59	912.5	912.59	962.59
	0.7	934.36	934.98	953.83	934.36	934.98	953.83	934.36	934.98	953.83	934.36	934.98	954.92
	0.8	956.21	957.36	976.81	956.21	957.36	977.04	956.21	957.36	976.81	956.21	957.36	977.04
	0.9	978.07	979.75	999.25	978.07	979.75	999.56	978.07	979.75	999.25	978.07	979.75	999.56
	1	999.92	1002.13	1002.13	999.92	1002.13	1002.13	999.92	1002.13	1002.13	999.92	1002.13	1002.13
G2 (320)	0.6	1406.52	1411.44	1462.5	1407.53	1411.44	1462.5	1407.53	1411.44	1451.58	1407.53	1411.44	1451.58
	0.7	1440.3	1446.63	1498.86	1441.21	1446.63	1498.86	1441.21	1446.63	1487.93	1441.21	1446.63	1487.93
	0.8	1474.08	1481.81	1519.03	1474.88	1481.81	1519.03	1474.88	1481.81	1527.16	1474.88	1481.81	1527.16

Continued on next page

Table B.1: – *Continued from previous page*

# (n)	α	C1			C2			M1			M2		
		CR	LR	LC	CR	LR	LC	CR	LR	LC	CR	LR	LC
	0.9	1507.86	1516.99	1585.91	1508.56	1516.99	1585.91	1508.56	1516.99	1585.91	1508.56	1516.99	1585.91
	1	1541.64	1552.17	1552.17	1542.23	1552.17	1552.17	1542.23	1552.17	1552.17	1542.23	1552.17	1552.17
G3 (400)	0.6	1970.04	1981.8	1994.77	1977.75	1980.82	1990.78	1977.7	1982.52	1994.77	1977.7	1982.52	1994.77
	0.7	2018.54	2032.47	2045.46	2026.23	2031.5	2041.31	2026.2	2033.17	2045.46	2026.2	2033.17	2045.46
	0.8	2067.04	2083.14	2101.92	2074.72	2082.18	2101.92	2074.7	2083.82	2099.93	2074.7	2083.82	2100.32
	0.9	2115.54	2133.81	2152.48	2123.21	2132.85	2151.26	2123.19	2134.47	2151.22	2123.19	2134.47	2151.17
	1	2164.05	2184.48	2184.48	2171.69	2183.53	2183.53	2171.69	2185.12	2185.12	2171.69	2185.12	2185.12
G4 (480)	0.6	2438.69	2481.12	2474.28	2446.14	2489.85	2508.03	2490.6	2555.52	2475.77	2490.75	2554.85	2514.68
	0.7	2496.03	2542.48	2475.64	2503.7	2551.34	2479.45	2549.23	2617.74	2486.52	2549.34	2616.91	2504.11
	0.8	2553.37	2603.84	2537.74	2561.25	2612.83	2562.48	2607.85	2679.96	2530.78	2607.92	2678.96	2578.19
	0.9	2610.71	2665.21	2666.88	2618.81	2674.33	2694.33	2666.47	2742.18	2696.01	2666.51	2741.01	2698.33
	1	2668.04	2726.57	2726.57	2676.36	2735.82	2735.82	2725.1	2804.4	2804.4	2725.1	2803.07	2803.07
G5 (200)	0.6	1156.26	1150.52	1142.2	1155.75	1141.73	1114.95	1155.3	1146.34	1122.07	1155.3	1145.39	1122.14
	0.7	1182.69	1177.6	1133.57	1181.99	1168.64	1123.05	1181.7	1173.27	1122.91	1181.7	1172.39	1122.91
	0.8	1209.13	1204.68	1160.31	1208.22	1195.54	1149.46	1208.11	1200.21	1149.36	1208.11	1199.39	1149.36
	0.9	1235.57	1231.76	1200.95	1234.45	1222.45	1209.63	1234.51	1227.14	1197.78	1234.51	1226.39	1201.95
	1	1262.01	1258.84	1258.84	1260.69	1249.36	1249.36	1260.92	1254.08	1254.08	1260.92	1253.38	1253.38
G6 (280)	0.6	1462.92	1460.17	1477.05	1462.92	1457.25	1477.45	1464.65	1452.9	1476.03	1464.65	1459.56	1476.03
	0.7	1497.34	1496.54	1506.04	1497.34	1493.52	1501.04	1499.48	1488.87	1530.85	1499.48	1495.76	1501.69
	0.8	1531.76	1532.92	1542.43	1531.76	1529.79	1537.16	1534.31	1524.85	1567.56	1534.31	1531.96	1537.46

Continued on next page

TABLE B.1: – *Continued from previous page*

# (n)	α	C1			C2			M1			M2		
		CR	LR	LC	CR	LR	LC	CR	LR	LC	CR	LR	LC
	0.9	1566.18	1569.29	1578.83	1566.18	1566.06	1573.28	1569.13	1560.82	1604.28	1569.13	1568.16	1573.23
	1	1600.61	1605.67	1605.67	1600.61	1602.33	1602.33	1603.96	1596.79	1596.79	1603.96	1604.36	1593.7
G7 (360)	0.6	1854.07	1865.03	1821.87	1854.07	1845.17	1840.48	1852.99	1862.69	1844.12	1852.99	1844.45	1844.12
	0.7	1899.35	1911.83	1868.39	1899.35	1892.04	1887.72	1898.02	1909.34	1891.53	1898.02	1891.28	1891.53
	0.8	1944.62	1958.64	1919.03	1944.62	1938.91	1927.22	1943.04	1955.99	1934.51	1943.04	1938.11	1934.51
	0.9	1989.9	2005.45	1980.64	1989.9	1985.78	1982.25	1988.06	2002.64	1975.91	1988.06	1984.94	1975.91
	1	2035.18	2052.26	2052.26	2035.18	2032.65	2032.65	2033.08	2049.29	2049.29	2033.08	2031.77	2031.77
G8 (440)	0.6	2040.19	2052.59	2070.47	2035.12	2049.8	2070.47	2046.8	2046.51	2070.47	2042.71	2046.85	2070.47
	0.7	2089.68	2103.56	2121.37	2084.46	2100.78	2120.86	2096.2	2097.41	2121.37	2092.37	2098.07	2120.86
	0.8	2139.18	2154.54	2146.7	2133.79	2151.76	2146.7	2145.61	2148.31	2143.54	2142.02	2149.29	2143.54
	0.9	2188.67	2205.51	2198.45	2183.12	2202.75	2198.45	2195.01	2199.21	2195.03	2191.68	2200.5	2195.03
	1	2238.17	2256.49	2256.49	2232.46	2253.73	2253.73	2244.42	2250.11	2250.11	2241.33	2251.72	2251.72
G9 (255)	0.6	100.31	98.91	99.33	100.31	99.55	98.97	100.31	99.85	99.2	100.31	98.93	98.97
	0.7	102.47	101.03	101.55	102.47	101.68	101.55	102.47	101.99	101.55	102.47	101.05	101.55
	0.8	104.62	103.14	103.68	104.62	103.82	103.68	104.62	104.12	103.68	104.62	103.17	103.68
	0.9	106.78	105.26	106.04	106.78	105.95	106.04	106.78	106.26	106.04	106.78	105.29	106.04
	1	108.93	107.38	107.38	108.93	108.09	108.09	108.93	108.39	108.39	108.93	107.41	107.41
G10 (323)	0.6	129.39	127.49	127.23	129.39	127.79	127.36	128.5	128.06	127.23	128.5	128.79	127.36
	0.7	132.16	130.21	130.04	132.16	130.53	130.08	131.26	130.81	130.04	131.26	131.55	130.08
	0.8	134.93	132.94	131.73	134.93	133.27	131.82	134.01	133.55	131.73	134.01	134.31	131.82

Continued on next page

TABLE B.1: – *Continued from previous page*

# (n)	α	C1			C2			M1			M2		
		CR	LR	LC	CR	LR	LC	CR	LR	LC	CR	LR	LC
	0.9	137.7	135.66	135.97	137.7	136.01	136.01	136.76	136.3	136.39	136.76	137.07	137.07
	1	140.47	138.38	138.38	140.47	138.75	138.75	139.52	139.05	139.05	139.52	139.83	139.83
G11 (399)	0.6	161.45	159.39	158.82	161.45	159.39	158.57	161.45	159.39	158.82	161.45	159.39	158.56
	0.7	164.91	162.81	162.2	164.91	162.8	161.97	164.91	162.81	162.2	164.91	162.8	161.97
	0.8	168.37	166.22	166.93	168.37	166.22	168.54	168.37	166.22	166.93	168.37	166.22	168.54
	0.9	171.83	169.63	169.61	171.83	169.63	168.54	171.83	169.63	169.61	171.83	169.63	168.6
	1	175.29	173.05	173.05	175.29	173.05	173.05	175.29	173.05	173.05	175.29	173.05	173.05
G12 (483)	0.6	192.53	192.46	192.38	192.53	191.11	193.3	193.05	192.39	193.08	193.05	191.75	193.3
	0.7	196.62	196.58	196.24	196.62	195.22	196.89	197.15	196.5	196.24	197.15	195.84	196.89
	0.8	200.72	200.7	202.68	200.72	199.34	200.77	201.25	200.61	202.68	201.25	199.93	202.68
	0.9	204.81	204.83	204.82	204.81	203.45	205.21	205.35	204.72	206.37	205.35	204.02	206.25
	1	208.91	208.95	208.95	208.91	207.56	207.56	209.45	208.83	208.83	209.45	208.11	208.11
G13 (252)	0.6	142.51	143.09	143.68	142.51	143.83	143.27	142.51	143.09	143.64	142.51	143.83	143.12
	0.7	145.63	146.21	148.32	145.63	146.95	148.9	145.63	146.21	149.08	145.63	146.95	148.4
	0.8	148.75	149.33	151.06	148.75	150.07	149.34	148.75	149.33	151.06	148.75	150.07	149.72
	0.9	151.87	152.46	152.4	151.87	153.18	153.2	151.87	152.46	152.4	151.87	153.18	153.2
	1	154.99	155.58	155.58	154.99	156.3	156.3	154.99	155.58	155.58	154.99	156.3	156.3
G14 (320)	0.6	181.75	181.1	182.04	181.75	181.5	182.65	181.75	181.1	182.68	181.75	181.58	182.31
	0.7	185.71	185.05	187.21	185.71	185.47	187.46	185.71	185.05	187.21	185.71	185.56	187.49
	0.8	189.66	189	189.63	189.66	189.45	188.55	189.66	189	189.63	189.66	189.53	189.78

Continued on next page

Table B.1: – *Continued from previous page*

# (n)	α	C1			C2			M1			M2		
		CR	LR	LC	CR	LR	LC	CR	LR	LC	CR	LR	LC
	0.9	193.62	192.95	193.06	193.62	193.42	193.62	193.62	192.95	193.06	193.62	193.51	193.71
	1	197.58	196.9	196.9	197.58	197.4	197.4	197.58	196.9	196.9	197.58	197.48	197.48
G15 (396)	0.6	225.21	224.95	226.46	225.21	225.44	226.44	225.21	224.56	226.46	225.21	225.23	226.4
	0.7	230.07	229.81	230.62	230.07	230.33	230.81	230.07	229.41	230.62	230.07	230.09	230.81
	0.8	234.94	234.67	236.77	234.94	235.22	237.26	234.94	234.27	236.66	234.94	234.96	237.26
	0.9	239.8	239.54	241.96	239.8	240.12	240.52	239.8	239.12	241.84	239.8	239.82	240.52
	1	244.66	244.4	244.4	244.66	245.01	245.01	244.66	243.98	243.98	244.66	244.68	244.68
G16 (480)	0.6	274.14	270.82	274.16	274.14	271.41	273.98	274.14	270.61	274.16	274.14	272.6	274.07
	0.7	280.05	276.69	280.5	280.05	277.29	279.32	280.05	276.48	280.5	280.05	278.5	278.92
	0.8	285.96	282.57	286.26	285.96	283.18	284.39	285.96	282.34	286.26	285.96	284.39	284.52
	0.9	291.87	288.45	291.32	291.87	289.07	289.4	291.87	288.21	291.82	291.87	290.29	288.86
	1	297.78	294.33	294.33	297.78	294.96	294.96	297.78	294.08	294.08	297.78	296.18	296.18
G17 (240)	0.6	121.59	120.98	119.89	121.59	121.79	119.42	121.57	121.23	119.89	121.25	121.79	119.27
	0.7	124.25	123.65	122.64	124.25	124.47	123.33	124.23	123.92	122.64	123.91	124.47	123.12
	0.8	126.91	126.31	126.44	126.91	127.15	126.1	126.88	126.6	126.44	126.57	127.15	126.1
	0.9	129.57	128.98	129.1	129.57	129.83	128.76	129.54	129.28	129.1	129.24	129.83	128.76
	1	132.23	131.65	131.65	132.23	132.51	132.51	132.19	131.97	131.97	131.9	132.51	132.51
G18 (300)	0.6	168.72	169.74	170.05	168.6	169.63	168.7	168.6	169.74	169.86	168.6	169.63	168.86
	0.7	172.35	173.42	173.42	172.23	173.3	172.5	172.23	173.42	173.42	172.23	173.3	172.76
	0.8	175.98	177.09	179.66	175.86	176.98	179.02	175.86	177.09	179.66	175.86	176.98	179.02

Continued on next page

Table B.1: – Continued from previous page

# (n)	α	C1			C2			M1			M2		
		CR	LR	LC	CR	LR	LC	CR	LR	LC	CR	LR	LC
	0.9	179.61	180.77	183.4	179.49	180.66	182.75	179.49	180.77	183.4	179.49	180.66	182.75
	1	183.23	184.45	184.45	183.12	184.34	184.34	183.12	184.45	184.45	183.12	184.34	184.34
G19 (360)	0.6	231.11	232.68	230.66	231.74	233.19	230.7	231.81	232.02	230.61	231.75	233.56	231.01
	0.7	236.14	237.79	238.55	236.79	238.31	238.91	236.87	237.11	239.09	236.81	238.69	239.41
	0.8	241.18	242.9	243.61	241.85	243.42	244.02	241.93	242.2	244.17	241.86	243.82	244.54
	0.9	246.21	248.01	248.37	246.9	248.54	249.68	246.99	247.29	248.59	246.92	248.94	249.77
	1	251.24	253.12	253.12	251.95	253.65	253.65	252.05	252.38	252.38	251.98	254.07	254.07
G20 (420)	0.6	310.24	309.99	313.25	310.1	308.77	312.95	309.4	310.99	313.25	309.24	310.23	312.95
	0.7	316.93	316.72	315.66	316.78	315.5	315.2	316.06	317.73	315.46	315.91	316.98	315
	0.8	323.62	323.44	322.33	323.47	322.23	321.89	322.73	324.46	322.12	322.57	323.73	321.68
	0.9	330.32	330.17	329	330.16	328.96	328.57	329.39	331.2	328.79	329.24	330.48	328.36
	1	337.01	336.9	336.9	336.85	335.68	335.68	336.06	337.94	337.94	335.9	337.23	337.23
						Large							
L1 (560)	0.6	3024.11	2951.63	2949.28	3025.58	2943.6	2944.62	3025.58	2946.6	2958.86	3025.58	2972.52	2957.97
	0.7	3097.93	3027.37	3075	3099.64	3019.59	3075	3099.64	3021.94	3074.22	3099.64	3048.85	3074.22
	0.8	3171.75	3103.11	3150.88	3173.69	3095.59	3150.88	3173.69	3097.27	3150.3	3173.69	3125.18	3150.3
	0.9	3245.58	3178.84	3220.68	3247.74	3171.58	3220.68	3247.74	3172.6	3217.9	3247.74	3201.51	3217.94
	1	3319.4	3254.58	3254.58	3321.79	3247.57	3247.57	3321.79	3247.93	3247.93	3321.79	3277.84	3277.84
L2 (600)	0.6	2461.01	2459.66	2442.73	2461.01	2460	2442.73	2461.01	2459.66	2442.73	2461.01	2460	2442.73
	0.7	2521.07	2521.01	2506.3	2521.07	2521.54	2505.59	2521.07	2521.01	2507.47	2521.07	2521.54	2505.81

Continued on next page

TABLE B.1: – *Continued from previous page*

# (n)	α	C1			C2			M1			M2		
		CR	LR	LC	CR	LR	LC	CR	LR	LC	CR	LR	LC
	0.8	2581.13	2582.36	2568.29	2581.13	2583.08	2567.61	2581.13	2582.36	2569.52	2581.13	2583.08	2567.91
	0.9	2641.18	2643.71	2631.81	2641.18	2644.63	2632.35	2641.18	2643.71	2631.9	2641.18	2644.63	2630.68
	1	2701.24	2705.06	2705.06	2701.24	2706.17	2706.17	2701.24	2705.06	2705.06	2701.24	2706.17	2706.17
L3 (640)	0.6	3467.07	3454.51	3440.15	3432.77	3449.48	3430.64	3452.68	3428.77	3441.04	3452.68	3437.95	3440.93
	0.7	3551.5	3544.44	3594.61	3516.96	3539.21	3596.71	3537.25	3517.46	3569.69	3537.25	3527.46	3569.69
	0.8	3635.94	3634.37	3605.88	3601.16	3628.93	3605.88	3621.82	3606.15	3614.77	3621.82	3616.97	3616.48
	0.9	3720.38	3724.3	3674.26	3685.35	3718.65	3674.44	3706.39	3694.84	3667.9	3706.39	3706.48	3670.43
	1	3804.81	3814.23	3814.23	3769.55	3808.38	3808.38	3790.96	3783.53	3783.53	3790.96	3795.99	3795.99
L4 (720)	0.6	4084.52	4057.59	3944.82	4084.49	4027.34	3940.57	4102.88	4057.66	3939.06	4102.88	4053.58	3934.17
	0.7	4184.4	4160.96	4010.64	4184.38	4130.1	4012.03	4202.81	4160.94	4018.45	4202.81	4156.8	4018.45
	0.8	4284.29	4264.33	4131.83	4284.27	4232.86	4131.71	4302.75	4264.21	4146.9	4302.75	4260.01	4146.9
	0.9	4384.17	4367.71	4278.51	4384.16	4335.62	4265.23	4402.68	4367.49	4279.41	4402.68	4363.23	4272.73
	1	4484.05	4471.08	4471.08	4484.05	4438.38	4438.38	4502.62	4470.76	4470.76	4502.62	4466.45	4466.45
L5 (760)	0.6	2840.4	2822.97	2850.6	2840.4	2819.38	2855.34	2840.4	2821.03	2853.99	2840.4	2820.48	2853.42
	0.7	2910.57	2895.2	2924.2	2910.57	2891.64	2929.4	2910.57	2893.17	2927.78	2910.57	2892.79	2927.13
	0.8	2980.74	2967.43	2991.87	2980.74	2963.91	2991.68	2980.74	2965.32	2997.46	2980.74	2965.1	2992.91
	0.9	3050.91	3039.66	3065.22	3050.91	3036.17	3065	3050.91	3037.46	3070.88	3050.91	3037.4	3066.5
	1	3121.08	3111.88	3111.88	3121.08	3108.44	3108.44	3121.08	3109.6	3109.6	3121.08	3109.71	3109.71
L6 (800)	0.6	4462.22	4416.28	4288.95	4465.53	4354.48	4288.95	4446.27	4402.39	4295.44	4446.29	4389.15	4295.03
	0.7	4571.75	4527.95	4355.8	4575.35	4466.17	4353.1	4555.81	4514.39	4355.64	4555.75	4502.11	4355.64

Continued on next page

TABLE B.1: – *Continued from previous page*

# (n)	α	C1			C2			M1			M2		
		CR	LR	LC	CR	LR	LC	CR	LR	LC	CR	LR	LC
	0.8	4681.28	4639.63	4687.58	4685.17	4577.87	4686.53	4665.36	4626.4	4660.47	4665.22	4615.06	4662.2
	0.9	4790.81	4751.31	4714.54	4794.99	4689.56	4711.66	4774.91	4738.4	4677.26	4774.68	4728.02	4677.26
	1	4900.33	4862.99	4862.99	4904.8	4801.26	4801.26	4884.46	4850.41	4850.41	4884.15	4840.97	4840.97
L7 (840)	0.6	2927.1	2921.21	2890.58	2927.1	2916.29	2890.74	2930.96	2928.05	2892.12	2930.96	2947.07	2892.89
	0.7	2998.12	2995.15	2984.9	2998.12	2990.4	2984.9	3002.08	3002.52	2984.98	3002.08	3021.55	2984.98
	0.8	3069.15	3069.09	3046.94	3069.15	3064.52	3047.26	3073.19	3076.99	3046.94	3073.19	3096.02	3047.26
	0.9	3140.17	3143.03	3146.04	3140.17	3138.63	3146.04	3144.31	3151.46	3146.04	3144.31	3170.5	3146.04
	1	3211.2	3216.97	3216.97	3211.2	3212.74	3212.74	3215.43	3225.93	3225.93	3215.43	3244.98	3232.91
L8 (880)	0.6	4938.67	4895.1	4818.13	4932.8	4849.37	4851.19	4927.41	4857.05	4834.17	4901.12	4845.42	4817.04
	0.7	5061.56	5020.03	5028.53	5055.53	4973.3	5022.92	5050.31	4981.05	5041.79	5023.45	4969.29	5030.43
	0.8	5184.46	5144.95	5135.57	5178.27	5097.23	5094.57	5173.21	5105.05	5138.42	5145.77	5093.17	5095.59
	0.9	5307.35	5269.87	5282.59	5301.01	5221.17	5206.63	5296.11	5229.04	5257.62	5268.1	5217.04	5159.83
	1	5430.24	5394.8	5394.8	5423.75	5345.1	5345.1	5419.01	5353.04	5353.04	5390.43	5340.91	5332.07
L9 (960)	0.6	5411.76	5342.26	5259.56	5422.37	5387.37	5199.08	5421.2	5414.54	5293.46	5420.69	5407.05	5221.87
	0.7	5547.23	5482.86	5232.23	5557.11	5529.15	5254.06	5556.32	5557.48	5256.89	5556.34	5549.84	5293.34
	0.8	5682.69	5623.46	5627.95	5691.86	5670.92	5634.14	5691.45	5700.41	5659.11	5691.99	5692.62	5659.11
	0.9	5818.16	5764.05	5808.44	5826.61	5812.7	5743.91	5826.57	5843.35	5797.38	5827.65	5835.41	5750.07
	1	5953.62	5904.65	5904.65	5961.36	5954.48	5954.48	5961.7	5986.29	5986.29	5963.3	5978.19	5963.47
L10 (1040)	0.6	5887.38	5843.19	5805.12	5862.39	5823.42	5802.38	5810.07	5804.71	5804.37	5814.09	5840.19	5802.05
	0.7	6035.44	5994.96	5962.82	6009.24	5975	5962.82	5955.99	5955.44	5969.23	5960.42	5991.92	5969.23

Continued on next page

Table B.1: – *Continued from previous page*

# (n)	α	C1			C2			M1			M2		
		CR	LR	LC	CR	LR	LC	CR	LR	LC	CR	LR	LC
	0.8	6183.5	6146.72	6141.56	6156.09	6126.58	6131.24	6101.91	6106.17	6142.2	6106.74	6143.64	6133.17
	0.9	6331.56	6298.48	6401.81	6302.93	6278.16	6401.81	6247.84	6256.89	6363.14	6253.06	6295.37	6363.14
	1	6479.62	6450.24	6450.24	6449.78	6429.74	6429.74	6393.76	6407.62	6407.62	6399.39	6447.09	6447.09
L11 (1120)	0.6	6368.44	6404.92	6189.79	6350.83	6390.7	6211.04	6371.27	6349.16	6226.97	6350.77	6369.27	6209.62
	0.7	6532.41	6574.61	6372.01	6513.87	6559.88	6376.87	6535.62	6517.43	6391.33	6513.93	6538.23	6391.33
	0.8	6696.38	6744.3	6775.63	6676.91	6729.06	6737.45	6699.96	6685.7	6776.74	6677.09	6707.19	6736.32
	0.9	6860.35	6913.99	6854.67	6839.96	6898.24	6850.42	6864.31	6853.96	6827.69	6840.26	6876.15	6848.68
	1	7024.32	7083.68	7083.68	7003	7067.41	7067.41	7028.66	7022.23	7022.23	7003.42	7045.11	7045.11
L12 (1200)	0.6	6784.51	6833.95	6817.91	6768.88	6766.9	6779.03	6776.41	6851.92	6818.36	6770.91	6853.55	6756.2
	0.7	6960.36	7019.03	7026.38	6944.5	6950.31	7007.33	6952.27	7037.73	7017.8	6946.59	7039.4	6993.25
	0.8	7136.2	7204.12	7187.39	7120.12	7133.71	7183.64	7128.12	7223.55	7227.73	7122.27	7225.25	7195.64
	0.9	7312.05	7389.2	7428.14	7295.74	7317.12	7438.96	7303.98	7409.36	7486.63	7297.94	7411.1	7476.21
	1	7487.9	7574.29	7574.29	7471.36	7500.53	7500.53	7479.84	7595.17	7595.17	7473.62	7596.95	7596.95

Bibliography

Apaydin, O. and Gonullu, M. T. (2008). Emission control with route optimization in solid waste collection process: A case study. *Sadhana* 33(2), 71–82.

Aras, N., Aksen, D., and Tuğrul Tekin, M. (2011). Selective multi-depot vehicle routing problem with pricing. *Transportation Research Part C: Emerging Technologies* 19(5), 866–884.

Barth, M., Younglove, T., and Scora, G. (2005). *Development of a heavy-duty diesel modal emissions and fuel consumption model*. California PATH research report. California PATH Program, Institute of Transportation Studies, University of California at Berkeley.

Bauer, J., Bektaş, T., and Crainic, T. G. (2009). Minimizing greenhouse gas emissions in intermodal freight transport: An application to rail service design. *Journal of the Operational Research Society* 61(3), 530–542.

Bektaş, T. and Laporte, G. (2011). The Pollution-Routing Problem. *Transportation Research Part B: Methodological* 45, 1232–1250.

Bing, X., de Keizer, M., Bloemhof-Ruwaard, J. M., and van der Vorst, J. G. (2014). Vehicle routing for the eco-efficient collection of household plastic waste. *Waste Management* 34(4), 719–729.

Björklund, M. (2011). Influence from the business environment on environmental purchasing — Drivers and hinders of purchasing green transportation services. *Journal of Purchasing and Supply Management* 17(1), 11–22.

Bula, G. A., Prodhon, C., Gonzalez, F. A., Afsar, H. M., and Velasco, N. (2017). Variable neighborhood search to solve the vehicle routing problem for hazardous materials transportation. *Journal of Hazardous Materials* 324, 472–480.

Çatay, B. (2010). A new saving-based ant algorithm for the vehicle routing problem with simultaneous pickup and delivery. *Expert Systems with Applications* 37(10), 6809–6817.

Chen, P., Huang, H.-k., and Dong, X.-Y. (2010). Iterated variable neighborhood descent algorithm for the capacitated vehicle routing problem. *Expert Systems with Applications* 37(2), 1620–1627.

CHLO (2017). *Data sets - Routing Data Sets - VRP*. Centre for Logistics and Heuristic Optimisation, Kent Business School, University of Kent. [Online]. Available from: https://www.kent.ac.uk/kbs/research/research-centres/clho/datasets.html.

Christofides, N. and Eilon, S. (1969). An algorithm for the vehicle-dispatching problem. *Operational Research Quarterly* 20(3), 309–318.

Christofides, N., Mingozzi, A., and Toth, P. (1979). The vehicle routing problem. In: Christofides, N., Mingozzi, A., Toth, P., and Sandi, C. eds. *Combinatorial Optimization*. Chichester: John Wiley & Sons, 315–338.

Ćirović, G., Pamučar, D., and Božanić, D. (2014). Green logistic vehicle routing problem: Routing light delivery vehicles in urban areas using a neuro-fuzzy model. *Expert Systems with Applications* 41(9), 4245–4258.

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

Dantzig, G. B. and Ramser, J. H. (1959). The truck dispatching problem. *Management Science* 6(1), 80–91.

DECC (2015). *2013 UK Greenhouse gas emissions, final figures - Statistical release*. Department of Energy and Climate Change. [Online]. Available from: https://www.gov.uk/government/statistics/final-uk-emissions-estimates.

Dekker, R., Bloemhof, J., and Mallidis, I. (2012). Operations research for green logistics – An overview of aspects, issues, contributions and challenges. *European Journal of Operational Research* 219(3), 671–679.

Demir, E., Bektaş, T., and Laporte, G. (2011). A comparative analysis of several vehicle emission models for road freight transportation. *Transportation Research Part D: Transport and Environment* 16(5), 347–357.

Demir, E., Bektaş, T., and Laporte, G. (2012). An adaptive large neighborhood search heuristic for the Pollution-Routing Problem. *European Journal of Operational Research* 223(2), 346–359.

Demir, E., Bektaş, T., and Laporte, G. (2014a). A review of recent research on green road freight transportation. *European Journal of Operational Research* 237(3), 775–793.

Demir, E., Bektaş, T., and Laporte, G. (2014b). The bi-objective Pollution-Routing Problem. *European Journal of Operational Research* 232(3), 464–478.

Dethloff, J. (2001). Vehicle routing and reverse logistics: The vehicle routing problem with simultaneous delivery and pick-up. *OR Spektrum* 23(1), 79–96.

Erdoğan, S. and Miller-Hooks, E. (2012). A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review* 48(1), 100–114.

Fagerholt, K., Laporte, G., and Norstad, I (2009). Reducing fuel emissions by optimizing speed on shipping routes. *Journal of the Operational Research Society* 61(3), 523–529.

Felipe, Á., Ortuño, M. T., Righini, G., and Tirado, G. (2014). A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. *Transportation Research Part E: Logistics and Transportation Review* 71, 111–128.

Figliozzi, M. A. (2012). The time dependent vehicle routing problem with time windows: Benchmark problems, an efficient solution algorithm, and solution characteristics. *Transportation Research Part E: Logistics and Transportation Review* 48(3), 616–636.

Fisher, M. L. (1994). Optimal solution of vehicle routing problems using minimum K-trees. *Operations Research* 42(4), 626–641.

Franceschetti, A., Honhon, D., Van Woensel, T., Bektaş, T., and Laporte, G. (2013). The time-dependent pollution-routing problem. *Transportation Research Part B: Methodological* 56, 265–293.

Gaur, D. R., Mudgal, A., and Singh, R. R. (2013). Routing vehicles to minimize fuel consumption. *Operations Research Letters* 41(6), 576–580.

Ghannadpour, S. F., Noori, S., Tavakkoli-Moghaddam, R., and Ghoseiri, K. (2014). A multi-objective dynamic vehicle routing problem with fuzzy time windows: Model, solution and application. *Applied Soft Computing* 14, 504–527.

Ghoseiri, K. and Ghannadpour, S. F. (2010). Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm. *Applied Soft Computing* 10(4), 1096–1107.

Gillett, B. E. and Miller, L. R. (1974). A heuristic algorithm for the vehicle-dispatch problem. *Operations Research* 22(2), 340–349.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research* 13(5). Applications of Integer Programming, 533 –549.

Golden, B. L., Wasil, E. A., Kelly, J. P., and Chao, I.-M. (1998). The impact of metaheuristics on solving the vehicle routing problem: Algorithms, problem sets, and computational results. In: Crainic, T. G. and Laporte, G. eds. *Fleet Management and Logistics*. New York: Springer Science+Business Media, 33–56.

Govindan, K., Soleimani, H., and Kannan, D. (2014). Reverse logistics and closed-loop supply chain: A comprehensive review to explore the future. *European Journal of Operational Research* 240(3), 603–626.

Hickman, J., Hassel, D., Joumard, R., Samaras, Z., and Sorenson, S. (1999). *Methodology for calculating transport emissions and energy consumption: Deliverable 22 for the project MEET 6.* Tech. rep. SE/491/98. Crowthorne: Transport Research Laboratory.

Holland, J. H. (1992). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence.* United States of America: MIT press.

Jin, J., Crainic, T. G., and Løkketangen, A. (2012). A parallel multi-neighborhood cooperative tabu search for capacitated vehicle routing problems. *European Journal of Operational Research* 222(3), 441–451.

Jovanović, A. D., Pamučar, D. S., and Pejčić-Tarle, S. (2014). Green vehicle routing in urban zones – A neuro-fuzzy approach. *Expert Systems with Applications* 41(7), 3189–3203.

Jozefowiez, N., Semet, F., and Talbi, E.-G. (2008). Multi-objective vehicle routing problems. *European Journal of Operational Research* 189(2), 293–309.

Kara, I., Kara, B. Y., and Yetis, M. K. (2007). Energy minimizing vehicle routing problem. *Lecture Notes in Computer Science* 1, 62–71.

Koç, Ç., Bektaş, T., Jabali, O., and Laporte, G. (2014). The fleet size and mix pollution-routing problem. *Transportation Research Part B: Methodological* 70, 239–254.

Kontovas, C. A. (2014). The Green Ship Routing and Scheduling Problem (GSRSP): A conceptual approach. *Transportation Research Part D: Transport and Environment* 31, 61–69.

Kramer, R., Subramanian, A., Vidal, T., and Cabral, L. d.A. F. (2015). A matheuristic approach for the Pollution-Routing Problem. *European Journal of Operational Research* 243(2), 523–539.

Kuo, Y. (2010). Using simulated annealing to minimize fuel consumption for the time-dependent vehicle routing problem. *Computers & Industrial Engineering* 59(1), 157–165.

Kuo, Y. and Wang, C.-C. (2011). Optimizing the VRP by minimizing fuel consumption. *Management of Environmental Quality: An International Journal* 22(4), 440–450.

Kwon, Y.-J., Choi, Y.-J., and Lee, D.-H. (2013). Heterogeneous fixed fleet vehicle routing considering carbon emission. *Transportation Research Part D: Transport and Environment* 23, 81–89.

Laporte, G. (2007). What you should know about the vehicle routing problem. *Naval Research Logistics* 54(8), 811–819.

Laporte, G. and Semet, F. (2002). Classical heuristics for the capacitated VRP. In: Toth, P. and Vigo, D. eds. *The Vehicle Routing Problem*. Philadelphia: SIAM Monographs on Discrete Mathematics and Applications, 109–128.

Li, F., Golden, B., and Wasil, E. (2005). Very large-scale vehicle routing: New test problems, algorithms, and results. *Computers & Operations Research* 32(5), 1165–1179.

Lin, C., Choy, K. L., Ho, G. T. S., and Ng, T. W. (2014a). A genetic algorithm-based optimization model for supporting green transportation operations. *Expert Systems with Applications* 41(7), 3284–3296.

Lin, C., Choy, K. L., Ho, G. T. S., Chung, S. H., and Lam, H. Y. (2014b). Survey of green vehicle routing problem: Past and future trends. *Expert Systems with Applications* 41(4), 1118–1138.

Lin, S. (1965). Computer solutions of the traveling salesman problem. *Bell System Technical Journal* 44(10), 2245–2269.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics* 21(6), 1087–1092. eprint: 5744249209.

Mladenović, N. and Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research* 24(11), 1097–1100.

Oberscheider, M., Zazgornik, J., Henriksen, C. B., Gronalt, M., and Hirsch, P. (2013). Minimizing driving times and greenhouse gas emissions in timber transport with a near-exact solution approach. *Scandinavian Journal of Forest Research* 28(5), 493–506.

Osman, I. H. (1993). Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research* 41, 421–51.

Osman, I. H. and Laporte, G. (1996). Metaheuristics : A bibliography. *Annals of Operations Research* 63, 513–623.

Palmer, A. (2007). *The development of an integrated routing and carbon dioxide emissions model for goods vehicles*. School of Management, Cranfield University. PhD thesis.

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

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

Pisinger, D. and Ropke, S. (2007). A general heuristic for vehicle routing problems. *Computers & Operations Research* 34(8), 2403–2435.

Pradenas, L., Oportus, B., and Parada, V. (2013). Mitigation of greenhouse gas emissions in vehicle routing problems with backhauling. *Expert Systems with Applications* 40(8), 2985–2991.

Price, R., Thornton, S., and Nelson, S. (2007). *The social cost of carbon and the shadow price of carbon: What they are, and how to use them in economic appraisal in the UK*. Tech. rep. London: Economics Group, Department for Environment, Food and Rural Affairs (DEFRA).

Privé, J., Renaud, J., Boctor, F., and Laporte, G. (2005). Solving a vehicle-routing problem arising in soft-drink distribution. *Journal of the Operational Research Society* 57(9), 1045–1052.

Ramos, T. R. P., Gomes, M. I., and Barbosa-Póvoa, A. P. (2014a). Economic and environmental concerns in planning recyclable waste collection systems. *Transportation Research Part E: Logistics and Transportation Review* 62, 34–54.

Ramos, T. R. P., Gomes, M. I., and Barbosa-Póvoa, A. P. (2014b). Planning a sustainable reverse logistics system: Balancing costs with environmental and social concerns. *Omega* 48, 60–74.

Rodrigue, J.-P., Comtois, C., Slack, B., Anderson, W., Bowen, J., Dablanc, L., Ducruet, C., Hesse, M., Notteboom, T., and Shaw, S.-L. (2017). *The Geography of Transport Systems*. Department of Global Studies and Geography, Hofstra University. [Online]. Available from: http://people.hofstra.edu/geotrans.

Salhi, S. (2017). *Heuristic search: The emerging science of problem solving*. Chem, Switzerland: Palgrave Macmillan.

Salhi, S. and Rand, G. K. (1987). Improvements to vehicle routeing heuristics. *Journal of the Operational Research Society* 38(3), 293–295.

Salhi, S. and Sari, M. (1997). A multi-level composite heuristic for the multi-depot vehicle fleet mix problem. *European Journal of Operational Research* 103(1), 95–112.

Salhi, S., Wassan, N., and Hajarat, M. (2013). The fleet size and mix vehicle routing problem with backhauls: Formulation and set partitioning-based heuristics. *Transportation Research Part E: Logistics and Transportation Review* 56, 22–35.

Salimifard, K., Shahbandarzadeh, H., and Raeesi, R. (2012). Green transportation and the role of operation research. In: *2012 International Conference on Traffic and Transportation Engineering*. Vol. 26. Ictte, pp. 74–79.

Sbihi, A. and Eglese, R. W. (2009). Combinatorial optimization and green logistics. *Annals of Operations Research* 175(1), 159–175.

Soysal, M., Bloemhof-Ruwaard, J. M., and Bektaş, T. (2015). The time-dependent two-echelon capacitated vehicle routing problem with environmental considerations. *International Journal of Production Economics* 164, 366–378.

Soysal, M., Bloemhof-Ruwaard, J. M., and van der Vorst, J. G.A. J. (2014). Modelling food logistics networks with emission considerations: The case of an international beef supply chain. *International Journal of Production Economics* 152, 57–70.

Suzuki, Y. (2011). A new truck-routing approach for reducing fuel consumption and pollutants emission. *Transportation Research Part D: Transport and Environment* 16(1), 73–77.

Tajik, N., Tavakkoli-Moghaddam, R., Vahdani, B., and Meysam Mousavi, S. (2014). A robust optimization approach for pollution routing problem with pickup and delivery under uncertainty. *Journal of Manufacturing Systems* 33(2), 277–286.

Tarantilis, C. D. (2005). Solving the vehicle routing problem with adaptive memory programming methodology. *Computers & Operations Research* 32(9), 2309–2327.

Tiwari, A. and Chang, P.-C. (2015). A block recombination approach to solve green vehicle routing problem. *International Journal of Production Economics* 164, 379–387.

Toth, P. and Vigo, D. (2002a). An overview of vehicle routing problems. In: Toth, P. and Vigo, D. eds. *The Vehicle Routing Problem*. Philadelphia: SIAM Monographs on Discrete Mathematics and Applications, 1–23.

Toth, P. and Vigo, D. (2002b). *The vehicle routing problem*. Ed. by P. Toth and D. Vigo. Philadelphia: SIAM Monographs on Discrete Mathematics and Applications.

Ubeda, S., Arcelus, F. J., and Faulin, J. (2011). Green logistics at Eroski: A case study. *International Journal of Production Economics* 131(1), 44–51.

Wassan, N. and Nagy, G. (2014). Vehicle routing problem with deliveries and pick-ups: Modelling issues and meta-heuristics solution approaches. *International Journal of Transportation* 2(1), 95–110.

Xiao, Y., Zhao, Q., Kaku, I., and Xu, Y. (2012). Development of a fuel consumption optimization model for the capacitated vehicle routing problem. *Computers & Operations Research* 39(7), 1419–1431.

Xu, J. and Kelly, J. P. (1996). A network flow-based tabu search heuristic for the vehicle routing problem. *Transportation Science* 30(4), 379–393.

Yang, P.-Y., Tang, J.-F., Yu, Y., and Pei, J.-X. (2013). Minimizing carbon emissions through vehicle routing and scheduling in the shuttle service of picking up and delivering customers to the airport. *Acta Automatica Sinica* 39(4), 424–432.

Zachariadis, E. E. and Kiranoudis, C. T. (2010). A strategy for reducing the computational complexity of local search-based methods for the vehicle routing problem. *Computers & Operations Research* 37(12), 2089–2105.

Zsigraiova, Z., Semiao, V., and Beijoco, F. (2013). Operation costs and pollutant emissions reduction by definition of new collection scheduling and optimization of MSW collection routes using GIS. The case study of Barreiro, Portugal. *Waste Management* 33(4), 793–806.