Petal-shaped clustering for the Capacitated Vehicle Routing Problem

Frances Bruwer (1557997)

Supervisor: Dr. Joke Bührmann

A research report submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, in partial fulfilment of the requirements for the degree of Master of Science in Engineering.

Johannesburg, February 2018

Declaration

I declare that this research report is my own unaided work. It is being submitted for the Degree of Master of Science in Engineering to the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination to any other University.

_____

(Signature of Candidate)

_____ day of _____ , 20_____ in _____

Abstract

In this research report, k-medoid (petal-shaped) clustering is modelled and evaluated for the Capacitated Vehicle Routing Problem (CVRP). To determine routes, an existing metaheuristic, termed the Ruin and Recreate method, is applied to each generated cluster. Results are benchmarked to that of a well-known clustering method, k-means clustering. The performance of the methods is measured in terms of travel cost and distance travelled, which are well-known metrics for Vehicle Routing Problems (VRPs). The results show that k-medoid outperforms the benchmark method for most instances of the test datasets, although the CVRP without any predefined clusters still provide solutions that are closer to optimal. Clustering remains a reliable distribution management tool and reduces processing requirements of large scale CVRPs.

Acknowledgements

I would like to express my gratitude to the following people for their contribution to this research:

- Dr. Joke Bührmann, my supervisor, for her guidance and encouragement throughout this study.
- Dr. Ian Campbell, for initiating the research topic and introducing me to this field.
- *Open Door Logistics*, for creating an excellent optimisation software application.
- My family, colleagues, and friends, for their continuous encouragement and support.

Table of contents

List of figures

List of tables

List of acronyms

CLRP	Capacitated Location Routing Problem	p 41
CVRP	Capacitated Vehicle Routing Problem	p 8
GPS	Global Positioning System	p 28
ODL	Open Door Logistics	p 27
PAM	Partitioning around Medoids	p 9
R&R	Ruin and Recreate	p 18
TSP	Travelling Salesman Problem	p 16
VRP	Vehicle Routing Problem	p 8
VRPTW	Vehicle Routing Problem with Time-Windows	p 43

1. Introduction

1.1 Research background

As distribution networks are growing in size and complexity, logistics companies require more advanced decision-making methods to serve its customer network. The advent of e-commerce has drastically increased the importance of distribution problems, as any household or workplace can instantly convert into a point of demand.

The Vehicle Routing Problem (VRP) is a widely studied problem in the field of operations research, generally applied in a distribution network to reduce transportation costs and improve service quality. A classic Vehicle Routing Problem (VRP) consists of one depot or facility, a fleet of homogeneous vehicles, and a set of geographically distributed customers with known locations and demand sizes. The main objective of a VRP is to determine a set of optimal routes, while minimising the total cost of delivery.

The Capacitated Vehicle Routing Problem (CVRP) is a well-known VRP variation that takes into account that vehicles hold a specific uniform capacity that may not be exceeded. The CVRP has been solved using a wide range of optimisation techniques that involve exact, heuristic and metaheuristic methods.

A key issue for a distribution manager is not only to decide on the number of vehicles to be used, but also to specify which customers to group and assign to a specific vehicle, and what sequence to follow, so as to minimise the transportation costs (Dondo and Cerdá, 2007).

Customers within close proximity can be grouped together with relative ease and this is often an intuitive process for distribution managers. However, in the case of larger distribution network problems, more structured clustering methods are required to group and allocate customers to specific routes.

Cluster analysis is the formal study of methods and algorithms for grouping or clustering objects according to measured or perceived inherent characteristics or similarity (Jain, 2010). Many clustering methods have been developed over the years, including a range of hierarchical, iterative partitioning, graph-based and nearest neighbour methods. The direct application of clustering techniques to CVRPs has not been widely documented.

This study highlights a specific clustering technique for VRPs, termed k-medoid (petal-shaped) clustering, which will be introduced and evaluated. K-medoid clustering will be evaluated as a potential method to effectively solve the CVRP. After the clustering method has been applied, the routing sequence of the vehicles will be determined for each cluster using an existing routing algorithm. To measure the effectiveness and feasibility of k-medoid clustering in CVRPs, the results of the problem will be compared to a selected benchmark clustering method.

1.2 Research motivation

Organising data into sensible groupings is one of the most fundamental modes of understanding (Jain, 2010). The application of clustering in vehicle routing can assist the distribution manager and employees in running a customer network in an ordered and structured manner. In practice, customers do not prefer very "abstract looking" solutions that appear counter-intuitive (Schrimpf *et al.*, 2000). Ideal solutions should therefore be easy to implement and interpret, while keeping costs at a minimum.

Although CVRPs have extensively been studied and a large amount of literature exists on various routing techniques, little exists on different clustering methods applied to routing problems. Buhrmann (2015) observed the interesting petal-shaped distributions created by the k-medoids clustering method. Ryan *et al*. (1993) defined the k-medoids clustering method as the Partitioning around Medoids (PAM) clustering method and recommended that the impact of using such clustering methods in VRPs could be expanded on.

The k-medoid clustering method applied to the CVRP was selected, as no previous works could be found on the subject. However, some of the best-known VRP solutions have naturally demonstrated petal-shaped groupings. This presented an opportunity to assess the viability of the k-medoid method for vehicle routing, and to propose ideal conditions under which it should be used. In order to measure the effectivity of k-medoid clustering in CVRPs, a benchmark clustering method was required. K-means clustering was selected, as it is widely used in data science applications and an extensive amount of literature is available.

1.3 Research objectives

The purpose of this research report is to determine the feasibility of k-medoid clustering for a large scale CVRP.

The research objectives are defined as follows:

1. Investigate and model a k-medoid clustering method for a CVRP.
2. Compare results of k-medoid clustering to a selected benchmark clustering method.
3. Propose characteristics and conditions to use k-medoid clustering for CVRPs.

1.4 Data and limitations

- Only k-medoid clustering is modelled and analysed for the CVRP. Other clustering methods are summarised in the literature review. K-means clustering is used to benchmark and validate the model.
- The model is applied to three different test datasets, obtained from *ODL Studio* (ODL Studio, 2014). CVRP parameters, such as the vehicle capacity and fleet size, will be set as realistic as possible, based on existing CVRP data libraries.

1.5 Report layout

The remainder of the report is structured as follows:

Chapter 2 contains a literature review on VRPs, routing methods, cluster analysis and existing clustering solution methods. *Chapter 3* describes the research method employed and explains the model, data, research tools, and the method used to validate the model. *Chapter 4* presents the model results obtained and an analysis of the findings. *Chapter 5* concludes the report with final remarks and recommendations for future areas of study.

2. Literature review

The literature study is divided into three sections. Section (2.1) defines the CVRP and looks at existing exact, heuristic and metaheuristic methods. Section (2.2) discusses cluster analysis and provides an overview of clustering methods relevant for distribution problems. The clustering method of this study, k-medoid (petal-shaped) clustering, and the benchmark method, k-means clustering, are also described in more detail. The last section (2.3), concludes with an interpretation of the literature.

2.1 The Capacitated Vehicle Routing Problem (CVRP)

The vehicle routing problem (VRP) was first introduced by Dantzig and Ramser (1959), and transformed the field of operations research. Solving VRPs remains as relevant as it was 60 years ago, as distribution networks continuously increase in size and complexity.

In this section, the CVRP will be formulated as described in literature and solution approaches that have been developed over the years are discussed.

2.1.1 CVRP Formulation

A variety of problem formulations exist for the CVRP. Laporte (1992) provides a problem formulation that is often referred to in literature, named the *three-index vehicle flow formulation*:

- Let G = (V, A) be a graph where V is a set of vertices, representing the set I of i=1 to n customers that need to be served. The network's depot is represented by node 0.
- Each customer has a specific demand w_i that must be met and may only be visited once.
- The associated cost or distance to travel from node i to j is represented by c_{ij}. In the case where $c_{ij} = c_{ji}$ the CVRP is said to be symmetrical.
- The CVRP has a set of K vehicles with a homogeneous capacity D, or a specific associated capacity of D(k) per vehicle k, where k Є K.

- Let S be a subset of nodes with $S \subset I$, $|S|$ represents the number of nodes of the subset.
- The binary decision variable x_{ijk} specifies whether a specific route (i,j) is traversed by vehicle k or not.

Minimise: $$\sum_{k=1}^{K} \sum_{j=0}^{n} \sum_{i=0, i \neq j}^{n} c_{ij} x_{ijk} \quad \text{(i)}$$

Subject to:

$$\sum_{j \in V} \sum_{k \in K} x_{ijk} = 1 \qquad \forall\, i \in I \quad \text{(ii)}$$

$$\sum_{i \in V} \sum_{k \in K} x_{ijk} = 1 \qquad \forall\, j \in I \quad \text{(iii)}$$

$$\sum_{i \in I} \sum_{j \in V} w_i\, x_{ijk} \leq D_k \qquad \forall\, k \in K \quad \text{(iv)}$$

$$\sum_{i,j \in S} x_{ijk} \leq |S| - 1 \qquad S \subset V, |S| \geq 2, \forall\, k \in K \quad \text{(v)}$$

$$x_{ijk} \in \{0, 1\} \qquad \forall\, i,j \in I, k \in K \quad \text{(vi)}$$

The objective function is to minimise the sum of the travel and vehicle costs associated with the routes of the vehicles. *(i)* Each customer may only be visited once, and by one vehicle only, as defined by constraints *(ii)* and *(iii)*. Constraint set *(iv*) checks that the capacities of the vehicles are not exceeded. Cycles or sub-tours are eliminated by constraint set *(v)*. Lastly, constraint set *(vi)* defines the binary nature of the decision variables.

2.1.2 CVRP Solution methods

Since the VRP was introduced in the operations research field, extensive work has been devoted to the problem and many optimisation algorithms and heuristics have been developed as a result (Laporte, 2009). This is due to the economic value of VRP solutions in logistics management, as well as the complexity of the problem itself. The VRP remains to be recognised as one of the most challenging problems in the field of combinatorial optimisation (Moolman, Koen and v.d. Westhuizen, 2010) and new advances and insights to the problem are continuously brought to light.

Table 2.1 shows a brief history of the advancement of vehicle routing solution methods over the past six decades:

Decade	*Major discoveries*
1950s	VRP first formulated as an integer problem (Dantzig and Ramser, 1959). Small problems (10-20 customers) are solved.
1960s	Early route-building heuristics proposed (Clarke and Wright, 1964). 2-opt and 3-opt applied to VRP (Christofides and Eilon, 1969). Problems of 30-100 customers are solved.
1970s	Two-phase heuristics proposed. Computational efficiency becomes important. Some larger problems (100-1000 customers) are solved.
1980s	Development of exact methods for the VRP.
1990s - 2000s	New metaheuristic methods are applied to the VRP e.g. Simulated annealing and Tabu search.

Table 2.1. A brief history of VRP solution methods

Solution methods for vehicle routing are generally categorised as exact, heuristic, or metaheuristic methods.

Exact methods allow for the finding of optimal solutions in smaller problems, but are often too time-consuming when solving real-world and larger problems. *Heuristic methods* employ problem-specific focused searches to find good solutions, but can lead to local optima solutions. *Metaheuristics* involve high level search procedures with built-in mechanisms to avoid that local optima are found. *Metaheuristics* require longer processing time, but deliver better results than heuristic methods (Hilier and Lieberman, 2015).

The most well-known solution methods are described in these categories below:

2.1.2.1 Classical heuristics

a. Clarke & Wright savings algorithm

A heuristic algorithm was presented by Clarke and Wright in 1964 based on the concept of savings. The method aims to make saving improvements by merging single customer routes based on the comparison of route calculated savings (Caccetta et al., 2013). Also known as the greedy approach, the algorithm has been one of the most widely used routing methods to date.

b. Fisher and Jaikumar algorithm

A two-phase method was presented by Fisher and Jaikumar (1981) that begins by allocating customers to specific vehicles through solving an assignment problem. Thereafter a routing sequence is determined for every vehicle. At the time that the method was presented, it outperformed all other heuristics.

c. K-opt heuristics

The k-opt method is a tour improvement heuristic that starts with an existing routing plan and aims to make improvements by exchanging routes (Laporte *et al.*, 2000). Iterations continue and improvements are made until a local optimum is found. The variable k represents the number of links to be exchanged. As an example, when 2 route vertices are removed per iteration, the method is referred to as 2-opt.

d. Nearest neighbour method

The nearest neighbour method was one of the foremost algorithms used to solve the Travelling Salesman Problem (TSP). It starts at a random point, and continuously finds the shortest edge until all points have been marked as visited. The algorithm usually yields a short tour, but not necessarily the optimal one (Laporte, 1992).

e. Route-first, cluster-second algorithms

The routing sequence is firstly calculated for a distribution network and thereafter customers are grouped together. Route-first, cluster-second methods have generally been found to not yield effective results for VRPs (Laporte *et al.*, 2000).

2.1.2.2 Exact methods

a. Branch-and-bound methods

The branch-and-bound method (Land and Doig, 1960) is a constrained-based heuristic which finds solutions by applying temporary relaxations to the problem. It uses a divide and conquer strategy to divide a solution space into subproblems and then solves each problem individually (Laporte, 1992).

b. Dynamic programming

Dynamic programming was first proposed for VRPs by Christofides *et al*. (1971). It solves sub-sets of a problem by iteratively storing results in a table and comparing results with previous iterations to determine best routes. Repeated work is avoided by taking advantage of previously partially computed solutions (Laporte, 1992).

2.1.2.3 Metaheuristics

a. Ant colony optimisation (ACO)

The ACO technique is based on the analogy of the behaviour of ant colonies that lay their trails to find optimal paths to food sources. It was first defined by Dorigo et al. in 1996 and has since been successfully applied to solve large scale VRPs. The algorithm makes use of positive feedback and a greedy heuristic for rapid discovery of good solutions (Blum, 2005).

b. Genetic algorithms

Evolutionary or genetic algorithms imitate the way species advance and adapt in its environment. It works with a selection process that only allows the "fittest" solutions to become parents and generate offspring solutions and repeats itself until a defined acceptable level of fitness is reached (Mohammed et al., 2012).

c. Simulated annealing

Based on the analogy of the physical annealing or strengthening process of metal in thermal dynamics, simulated annealing presented a new perspective on traditional optimisation problems. It uses a random search heuristic that escapes finding the local optima by allowing larger jumps with the aim of finding the global optimum (Henderson et al., 2003).

d. Tabu search

Tabu search employs local searches to continuously optimise existing routing solutions. The method was first presented by Glover (1989). The heuristic constructs a sequence of solutions, and then executes improvement steps. A "tabu" list of previously found results is recorded to avoid repetitions or cycling (Blum and Roli, 2003).

2.1.2.4 The Ruin and Recreate (R&R) method

The routing software that was used in this study provided a useful base to execute, measure, and interpret results. The optimisation method that the software is based on, was examined by the author to gain understanding of its operating method.

The Ruin and Recreate (R&R) method was presented by Schrimpf *et al.* (2000). It employs a combination of heuristics and metaheuristics. The notion of simulated annealing is incorporated, with the effect that larger moves or jumps are made to find the optimum and to avoid getting trapped in local optima.

A new tactic was introduced to classic optimisation methods by the R&R method, whereby new solutions are found by removing sections of existing solutions and then rebuilding them.

This R&R concept can be explained as follows:

1. Specific destinations or customers are removed ("ruined") from the original configuration of customers to be serviced. These customers are usually selected based on a radial measurement from the central depot.
2. Customers that were removed previously are attempted to be added again ("recreated"). The best possible insertion of the customer is searched for i.e. the customer is added in the least expensive manner.
3. The most suitable vehicle is selected for the added customer that will imply the minimum cost. In the case of a capacity constraint, where the added customer cannot be added with current resources, an additional vehicle is added to the solution.
4. The new solution is either accepted or rejected based on a defined decision rule. If the solution is not accepted, steps (1) to (4) are repeated.

The R&R heuristic achieved record breaking results for a range of classical optimisation problems (Schrimpf *et al.*, 2000). All the VRP instances in the paper showed best published numerical results.

2.2 Clustering

2.2.1 Clustering analysis

Clustering analysis can be defined as the task of merging or dividing data points to form meaningful, useful groups. The objective of clustering analysis is to minimise the total dissimilarity between data points in clusters (Gore, 2000). This can be achieved by using a variety of advanced algorithms, depending on the application and defined criteria. However, in many cases the task of allocating items or data points together is an intuitive process (Everitt *et al.*, 2011).

As distribution networks increase in size, clustering becomes important to split customers into subsets. Arranging customers in homogenous groups is not only useful from an operational management point of view, but can also advise routing decisions. In large-scale distribution network problems, a successful VRP application should include both customer clustering and vehicle routing optimisation (Wang *et al.*, 2015).

The cluster-first, route-second heuristic was first defined by Fisher and Jaikumar (1981) to solve vehicle routing problems in a two-phase approach. In the first phase, customers are clustered into the same number of clusters as the number of vehicles to be used. Thereafter, the routing sequence is determined for every cluster. The routing of these clusters is the well-known traveling salesman problem (TSP).

The challenge is to apply clustering in such a way that the sum of route distances is collectively minimised in each cluster (Fisher and Jaikumar, 1981).

2.2.2 Clustering methods

Bührmann (2015) has classified clustering methods for geographical data into five main types:

1. Hierarchical methods
2. Iterative partitioning
3. Graph-based methods
4. Nearest neighbour methods
5. Density-based methods

These clustering categories have been summarised in *Table 2.2*:

Clustering method	Description	Clustering techniques
1. Hierarchical clustering	Customers are either merged or divided iteratively until stopping criteria is reached, such as a defined number of clusters.	- Single/complete linkage method - Average linkage method - Weighted average-linkage method - Centroid linkage method - Weighted centroid linkage method - Ward's method
2. Iterative partitioning	The method starts off with a number of existing seed points (customers). Customers that are closest to seed points will be allocated together. Cluster centroids/medoids/medians are repeatedly calculated and the goal is to minimise the sum of the distances between all customers and cluster centres.	- k-means method - k-medoid method - k-median method
3. Graph-based methods	Customer points are represented on a graph where points represent nodes and links are connections between nodes. Connections that are "inconsistent" are removed. The remaining connected customers represent clusters.	- MST graph-based method - RNG graph-based method - GG graph-based method
4. Nearest neighbour methods	All customer points' nearest neighbour points are calculated and added to the same clusters.	- k-near neighbours method - Mutual neighbourhood value
5. Density-based method	This method works on the basis that clusters can be viewed as areas that that are dense, and surrounded by areas that are low in density. Cluster centres are placed in regions where the most points are.	- k'th nearest neighbour - Hybrid clustering-based method

Table 2.2. Clustering methods for geographical data (Bührmann, 2015)

2.2.3 K-medoid clustering

The k-medoid method forms part of the *iterative partitioning* cluster method category. The method was first introduced by Kaufman and Rousseeuw (1990) as a variation of other partitioning methods. The main difference compared to other partitioning methods, such as the well-known k-means method, is that it works with medoids as opposed to centroids.

The medoid of a cluster can be defined as the most centrally located point in a cluster, or the point in the cluster where the average dissimilarity to all the other points in the cluster is at a minimum. (Kaufman and Rousseeuw, 1990)

As with other partitioning methods, there needs to be a starting solution for the problem. Points are then iteratively re-assigned as the best possible medoid is calculated for every cluster repeatedly.

The cluster shapes associated with the k-medoid method can be described as petal- or pie-shaped as can be seen in the example of results in *Figure 2.1*.

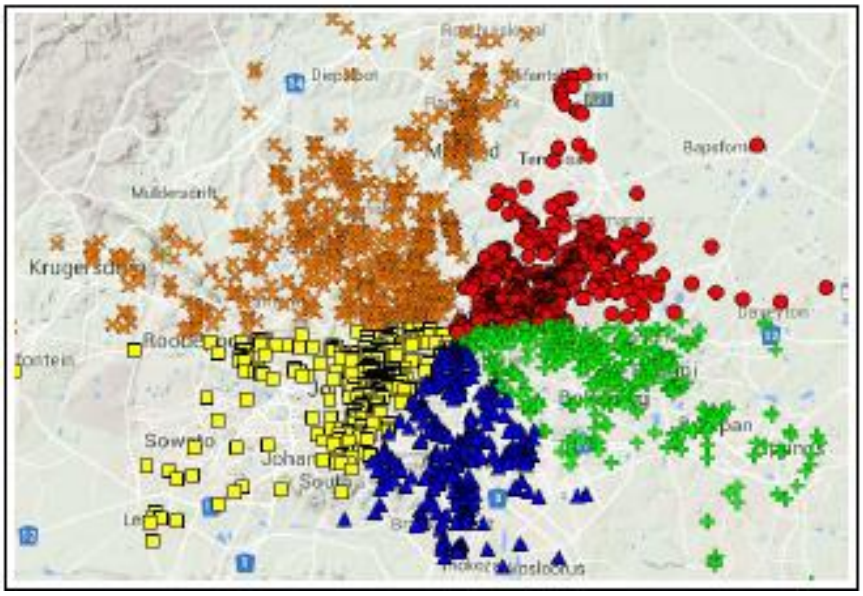

Figure 2.1. Example of k-medoid geographical clusters (Bührmann, 2015, p141)

K-medoid algorithm description

The k-medoid clustering method can be explained in the following high-level steps:

1. Choose a set of k medoids as initial seed points.

2. Calculate the dissimilarity between all points.

3. Allocate every point to its closest medoid.

4. Compare dissimilarities of all points in clusters.

5. Change points that lower total dissimilarity to become new medoids.

6. If any medoid has been re-allocated, go back to step 3.

Application in CVRPs

Ryan *et al.* (1993) observes that optimal solutions for many problems exhibit petal-shaped structures, and might be useful in finding good solutions for practical-sized problems.

However, there is limited literature available on the use of the k-medoid method or petal-shaped clusters for CVRPs. The effectivity of the method applied to vehicle routing and the type of datasets that would work well for the method remain to be assessed.

2.2.4 K-means clustering

The most well-known iterative clustering method, k-means, was selected as a benchmark method for the model in this study.

K-means clustering is a relatively simple method to implement. It has widespread applications and a large amount of literature is available. More recently, it has become a popular data exploration method for unsupervised learning problems, where similar characteristics are grouped in unlabelled data (Jain, 2010).

As with k-medoid clustering, the clustering process begins by randomly selecting an amount of seed points equal to k. Every customer is allocated to a cluster by associating it to its nearest mean. Potential distance savings is calculated for every customer associated to alternative clusters. A reassignment is executed if a potential distance savings can be made. Cluster centres

are recalculated after every reassignment, which become the new means of the clusters. Customer assignments and the recalculation of cluster centroids are repeated until no further distance savings can be made or convergence is reached.

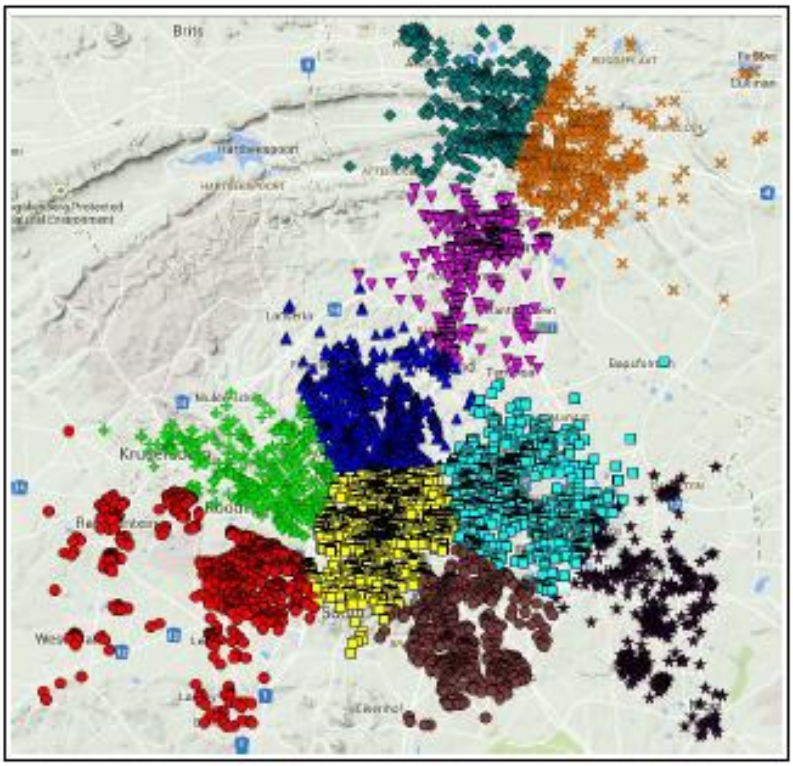

Figure 2.2. Example of k-mean geographical clusters (Bührmann, 2015, p145)

2.2.5 Determining the number of clusters (k)

Theoretically, there is no exact method for determining the appropriate number of clusters (k) of a specific dataset, and it is often an ambiguous process of trial and error (Salvador and Chan, 2003). No methods could be found specifically for distribution network problems. A trial and error approach was used in this study where a range of different number of clusters were tested to determine the optimal k. It remains an important problem to solve in cluster analysis and a few techniques have been suggested in literature. The most well-known methods are explained below:

a. The elbow point method

The number of clusters of a dataset is determined by plotting the average distances within a cluster to its centroid against the number of clusters. The turning point in the curve, where the average distance decreases significantly, indicates the number of clusters that should be selected (Salvador and Chan, 2003). *Figure 2.3.* demonstrates an example:

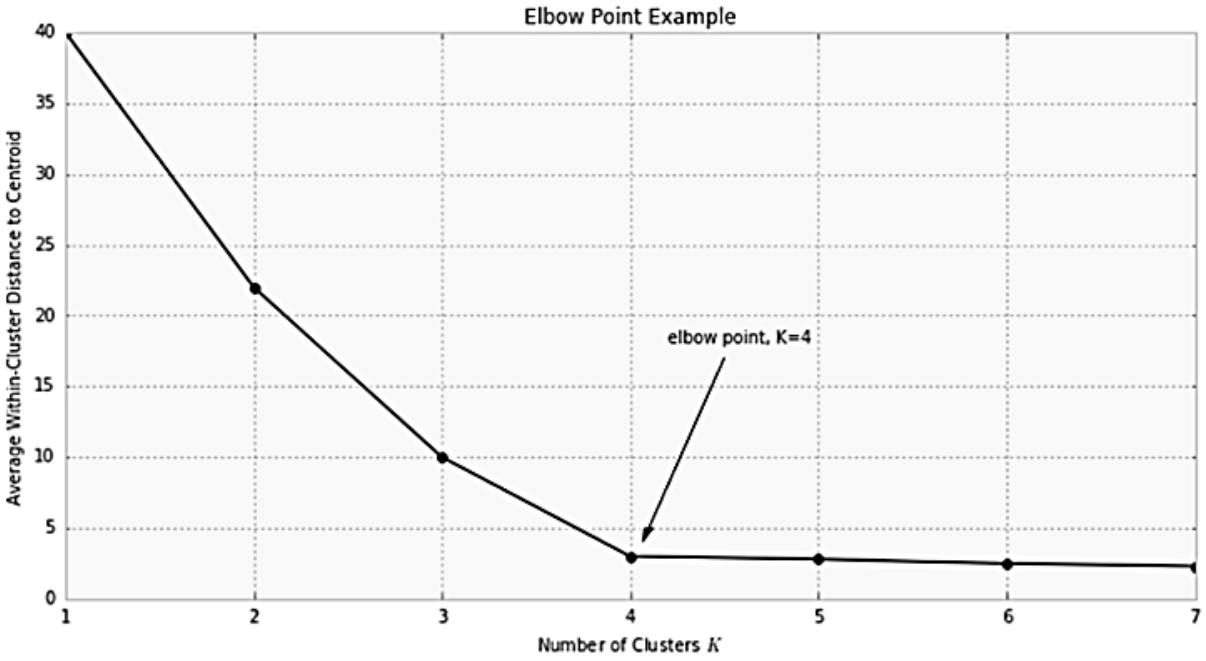

Figure 2.3. The elbow point of a clustered dataset (k=4)

b. The silhouette method

The level of fit of each data point is measured in its current cluster and compared to the level of fit to its neighbouring cluster. A silhouette measurement of 1 is considered as correct clustering and a measurement of close to -1 implies incorrect clustering. The optimal number of clusters is the one that generates the highest average silhouette over a range of possible values for k (Kaufman and Rousseeuw, 1990).

c. Cross-validation

In this technique, the dataset is partitioned into k subsets. One of these subsets is analysed as a training set, and validated by the average of all other subsets. All the subsets are then, in turn, analysed as training sets and compared to the average of the remaining subsets. The cluster number is selected such that a further increase of the number of clusters will only associate a minor change in results (Kohavi, 1995).

2.3 Literature review conclusion

A great amount of work has been dedicated to the topic of vehicle routing. The VRP inspired the development of numerous exact, heuristic and metaheuristic methods over the years. Due to its commercial and academic relevance, new insights to the VRP are still being pursued by the operational research community today.

Literature on clustering analysis was easily obtainable, but for diverse applications. Although a variety of studies on clustering methods were found, very few of those were aligned to the application of clustering contemplated in this report. Bührmann (2015) provided a comprehensive categorisation of clustering methods, specifically for distribution problems, that was used in this study. Furthermore, the determination of an appropriate number of clusters proved to be an important consideration in the process, and prompted investigation into techniques for determining the optimal value of k.

Overall, the application of clustering to solve VRPs remains to be expanded on. No examples could be found of k-medoid clustering applied to the CVRP.

3. Research method

The purpose of the designed method was to measure the effectiveness of k-medoids clustering for the CVRP. Although several algorithms and software applications are available to address the routing section of the problem, clustering tools for distribution network problems are scarce. Therefore, the clustering method had to be developed and tested by the author. The cluster results, along with other CVRP parameters, were then provided as input to an existing routing software package that determined the routes of the vehicles to the customer locations. Thus, a *cluster-first, route-second* approach was applied.

To ensure that unbiased observations could be made, datasets of different sizes and characteristics were modelled. The input parameters for the CVRP were selected based on existing benchmarked datasets. Results were measured by the total distance travelled by the fleet and total cost incurred.

The chapter begins with an overview of the method and the research tools used in this study. Thereafter, each step in the methodology depicted in *Figure 3.1* is discussed. A section on model validation concludes the chapter.

3.1 Methodology overview

The research method applied can be summarised in the following high-level steps:

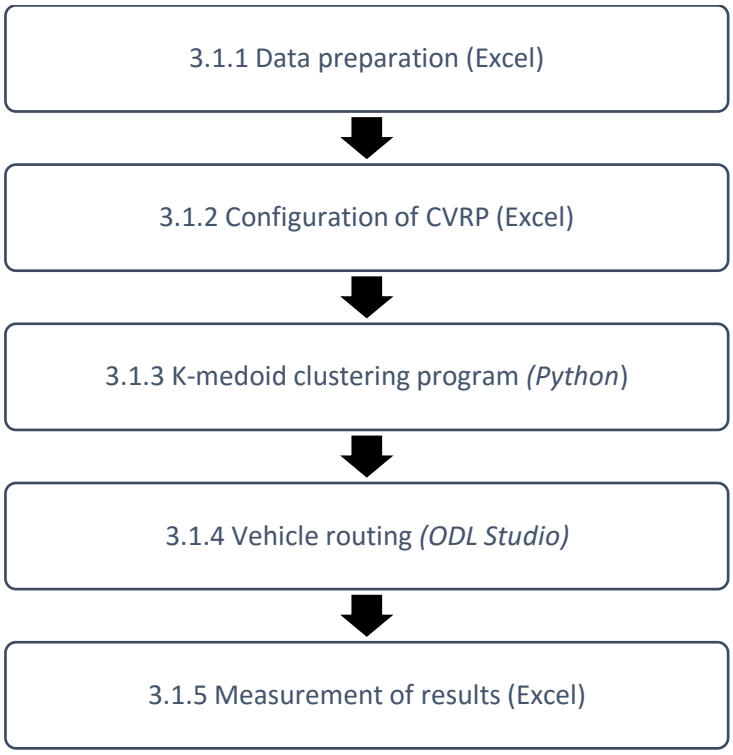

Figure 3.1. Research method overview

Research tools

The clustering section (3.1.3) of the model was coded in *Python*. *Python* is a widely used general-purpose programming language that can easily be expanded with supplementary modules such as matrix manipulation and the plotting of functions. An extensive library of existing functions and support was available. (Python Software Foundation, 2001)

For the routing section (3.1.4) of the model, an existing software package, *Open Door Logistics Studio (ODL Studio),* was used. *ODL studio* is an open source desktop application for the planning of non-real-time vehicle fleet routing and scheduling (ODL Studio, 2014). It is based on the R&R method described in *section 2.1.2.4*, which uses a combination of heuristics and metaheuristics to find optimal routes. The program also offers other useful features, such as the mapping of routes and integration with *Microsoft Excel*.

Data preparation and the analysis of results were done in *Microsoft Excel*. All tests were carried out on an *Intel Core i7/2.6 GHz* computer with *4 GB RAM* and *Windows 10* installed as an operating system.

3.1.1 Data preparation

The input data required for the model were as follows:

- Maximum capacity of vehicles
- Travel and fixed costs of vehicles
- Vehicle fleet size
- Customer demand at each location
- Geographical locations of customers

The maximum capacity of vehicles and fleet size were selected based on existing benchmarked CVRP problems. The demand of all customer locations was kept uniform.

Data format of geographical locations of customers

Multiple CVRP data libraries exist and the data structure of the libraries vary. Customer locations are listed in different coordinate formats. The decision was made to use datasets in *longitude and latitude decimal format*. This proved beneficial, as this allowed the results to be geographically mapped in the routing software. Having the calculated routes visualised in a real-world setting provided a more intuitive method of interpreting the results. Furthermore, this opens up the possibility for integration with GPS, which would permit scaling to real-world distribution problems.

In the *longitude and latitude decimal format*, the latitude is a positive value when a location is north of the equator and negative when south. Similarly, the longitude is positive when east of the prime meridian and negative when west (OSGB, 2010). As an example, all locations in South Africa have a negative latitude value and a positive longitude value.

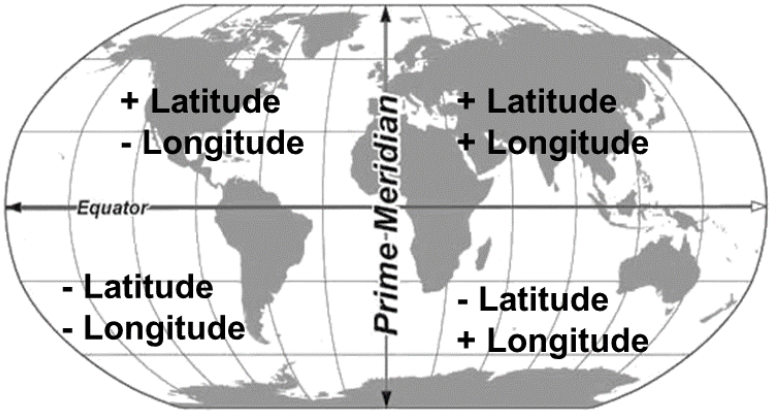

Figure 3.2. Locations format: Longitude and latitude decimal format

Three different datasets of varying sizes and locations were used to ensure that unbiased conclusions can be made at the analysis stage. The number of clusters were also varied per dataset to assess the effect of cluster sizes on the results.

The datasets were defined as follows:

- Dataset A: United Kingdom (size 100)
- Dataset B: Germany (size 500)
- Dataset C: Austria (size 2000)

All three datasets were sourced from *ODL Studio's* demo data library (ODL Studio, 2014). Each dataset consists of two tables that serve as input for the CVRP. Firstly, the customer locations with latitude and longitude values and demand at each customer, and secondly a table with parameters for the vehicle fleet. This includes the vehicle size, vehicle capacity, cost per kilometre and fixed cost per vehicle. Samples of the CVRP input tables are included in *Appendix C*.

3.1.2 Configuration of CVRP

A CVRP with a single central depot and many customers was modelled in this study.

The objective function of a classic CVRP is to calculate the set of routes that minimise the total cost of delivering to customers. This cost consists of vehicle fleet costs and travelling costs (distance travelled). The results of this study were similarly measured.

The following CVRP model constraints were adhered to:

- All customers' demands must be met
- All customers must be served exactly once
- All vehicles have a known capacity that may not be exceeded
- Every customer may only be served by one vehicle
- Each vehicle starts and ends its route at the central depot

The following assumptions relating to the CVRP were made in this study:

- All vehicles were identical and had a uniform capacity
- All customers had a uniform demand
- There were no delivery time windows specified
- The customer central depot is indicated in the dataset
- The total travel cost is calculated as the sum of fixed costs per vehicle and the total travel cost per kilometre of all trips.

3.1.3 K-medoid clustering program

The k-medoid clustering program developed in *Python* can be explained in the following steps:

i. Read customer location coordinates from *Excel*

ii. Calculate the distances between all customer locations to create a distance matrix using the Euclidean distance formula:

For points $p = (p_1, p_2, \ldots p_n)$ and $q = (q_1, q_2, \ldots q_n)$

$$d(p, q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

iii. Initialise an array of k medoids from the customer locations.

iv. Associate each data point to the closest medoid.

v. For each medoid and each non-medoid point, calculate the less costly alternative and update the medoids accordingly.

vi. Repeat the previous step until the best solution is reached.

v. Return the array of customer locations with solution cluster indices.

The clustering results were then plotted using the *matplotlib* function in Python. The example in *Figure 3.3* below shows the k-medoid clustering results of USA test data (ODL Studio, 2014) of 1000 customers with 6 clusters:

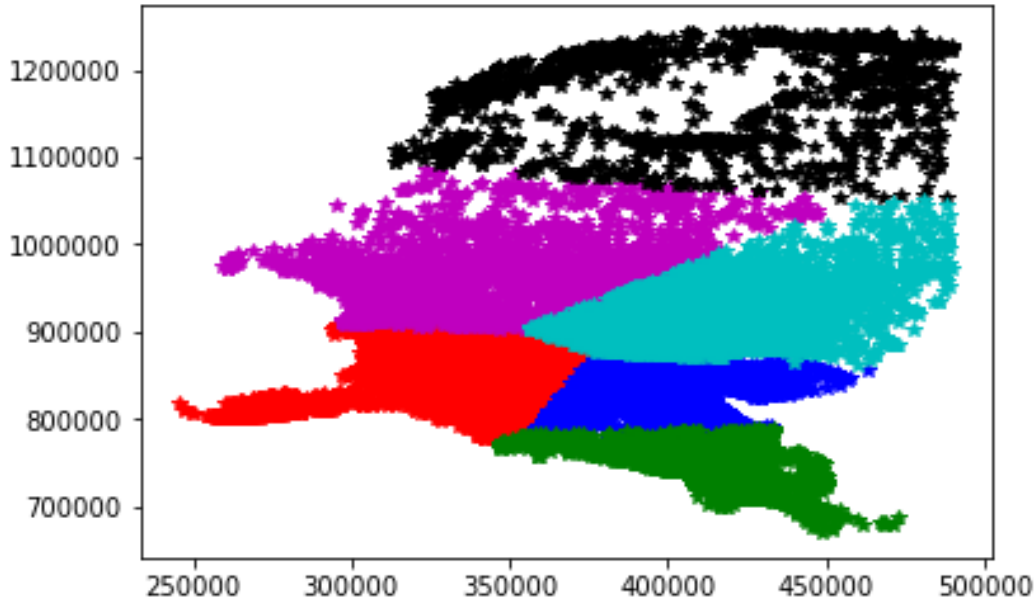

Figure 3.3. Example of k-medoid clustering results plotted in Python for USA test data with k= 5 (ODL Studio, 2014)

3.1.4 Vehicle routing

After the customer locations have been clustered, the routing section of the model was executed in *ODL Studio*.

The customer locations with their associated cluster indexes, as determined by the clustering program, were provided as input to the routing software. All other CVRP parameters, such as vehicle capacity and fleet size were also set up in *ODL Studio*.

The software makes use of a combination of heuristics and metaheuristics, referred to as the R&R approach to determine optimal routes (refer to *section 2.1.2.4*). The optimisation method uses Simulated Annealing with bold, large moves instead of smaller ones to find optimal solutions. By performing this type of change frequently in classic optimisation problems, Schrimpf *et al.* (2000) found that the R&R implementation achieved the best results for their datasets.

The solution of the routing optimisation was mapped using the geocoding mapping functionality of the software. The example in *Figure 3.4* below displays the routing results of a USA dataset with 200 customers grouped into five clusters (ODL Studio, 2014).

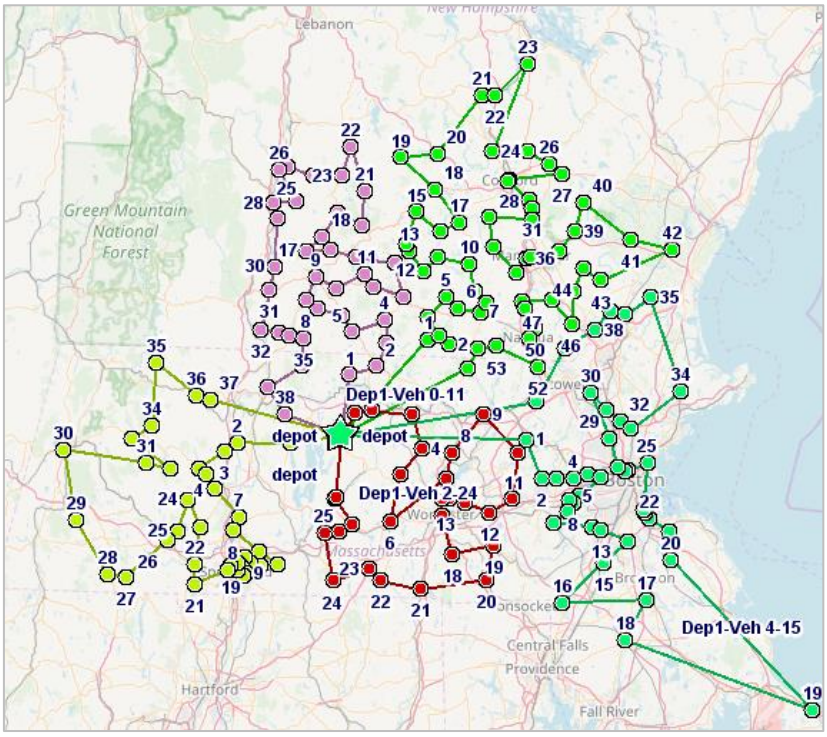

Figure 3.4. Geographical map example of routing solution: ODL Studio software

3.1.5 Measurement of results

The method explained in *section 3.1.1-3.1.5* was executed for different scenarios to view the effect on the final results. Three datasets of varying customer locations were used and the number of clusters (k) were incremented for each dataset.

For every iteration, the following performance metrics were captured in *Microsoft Excel*:

- Total travelling costs (combination of fixed and cost per kilometre)
- Total distance travelled
- Possible violation of constraints (e.g. unassigned customers or vehicle capacity violation)

Finally, the generated routing solution was analysed visually on the geographic map. The findings and analysis of the results are discussed in Chapter 4.

3.2 Model validation

The model was validated by selecting an existing, well-known clustering method to compare the results found in this study. The same input parameters were used for both instances and the routing was executed similarly for both clustering methods.

Bührmann (2015) defined a list of clustering methods in her research that were considered as benchmark methods in this study. It was decided to use k-means clustering as a benchmark as this is a relatively simple, widely-used clustering method that is significantly covered in literature. (See 2.2.4)

K-means is a popular clustering algorithm that has widespread application in data science, more recently particularly in machine learning problems. The k-means function is included in the *scikitlearn* module of *Python* (Pedregosa *et al.*, 2012) and could easily be imported into the model. This proved to be a strong benchmark method for this study.

4. Results and analysis

4.1 K-medoid computational results

The three different datasets were clustered by implementing the k-medoid method and the benchmark method, k-means, over a range of k-values. After the clustering results were imported to the routing software, route sequences were calculated, and the resulting CVRP performance metrics were recorded for every iteration.

The k-medoid method showed promising results compared to the benchmark method, k-means. In most instances, better results were obtained in terms of the total distance travelled and the total cost. The performance metrics and a visualisation of the generated k-medoid clusters are displayed below for each dataset.

4.1.1 Dataset A: United Kingdom (size=100)

Number of clusters	Travel cost		Distance travelled (km)	
	K-medoid	K-means	K-medoid	K-means
k=3	27.43	27.91	2043	2079
k=4	28.85	31.37	2149	2336
k=5	30.52	31.59	2273	2353
k=6	32.64	33.66	2431	2507

Table 4.1. Results of k-medoid and benchmark method for dataset A

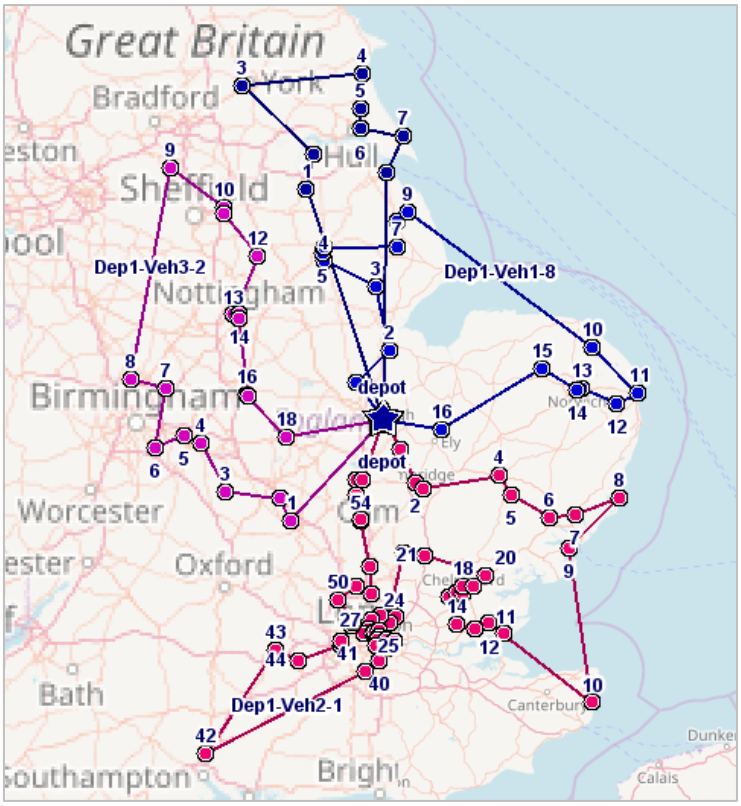

Figure 4.1. K-medoid clustering results for dataset A (UK, k=4)

4.1.2 Dataset B: Germany (size=500)

Number of clusters	Travel cost		Distance travelled (km)	
	K-medoid	K-means	K-medoid	K-means
k=5	121.51	126.90	9049	9450
k=6	124.48	125.02	9270	9311
k=7	125.59	126.92	9353	9452
k=8	131.06	132.11	9761	9839
k=9	132.59	138.13	9874	10288
k=10	139.05	144.59	10356	10768

Table 4.2. Results of k-medoid and benchmark method for dataset B

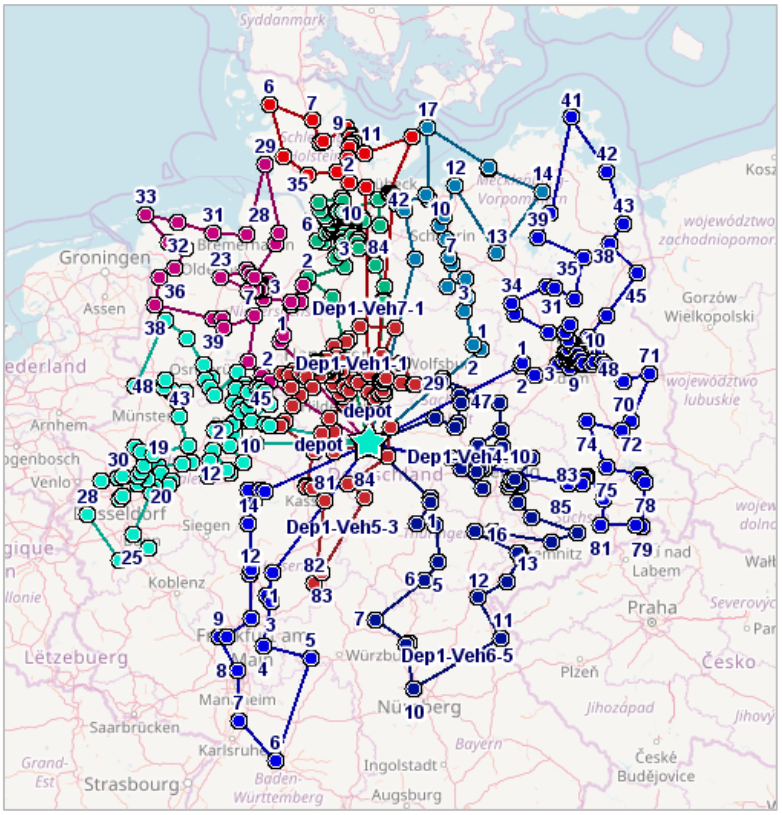

Figure 4.2. K-medoid clustering results for dataset B (Germany, k=5)

4.1.3 Dataset C: Austria (size=1500)

Number of clusters	Travel cost		Distance travelled (km)	
	K-medoid	K-means	K-medoid	K-means
k=12	57.68	58.49	4296	4356
k=13	57.24	58.16	4263	4331
k=14	58.28	58.40	4341	4349
k=15	58.69	59.58	4371	4437
k=16	60.95	60.73	4539	4523
k=17	59.34	61.68	4419	4593
k=18	59.83	59.05	4456	4398
k=19	61.99	62.45	4616	4651

Table 4.3. Results of k-medoid and benchmark method for dataset C

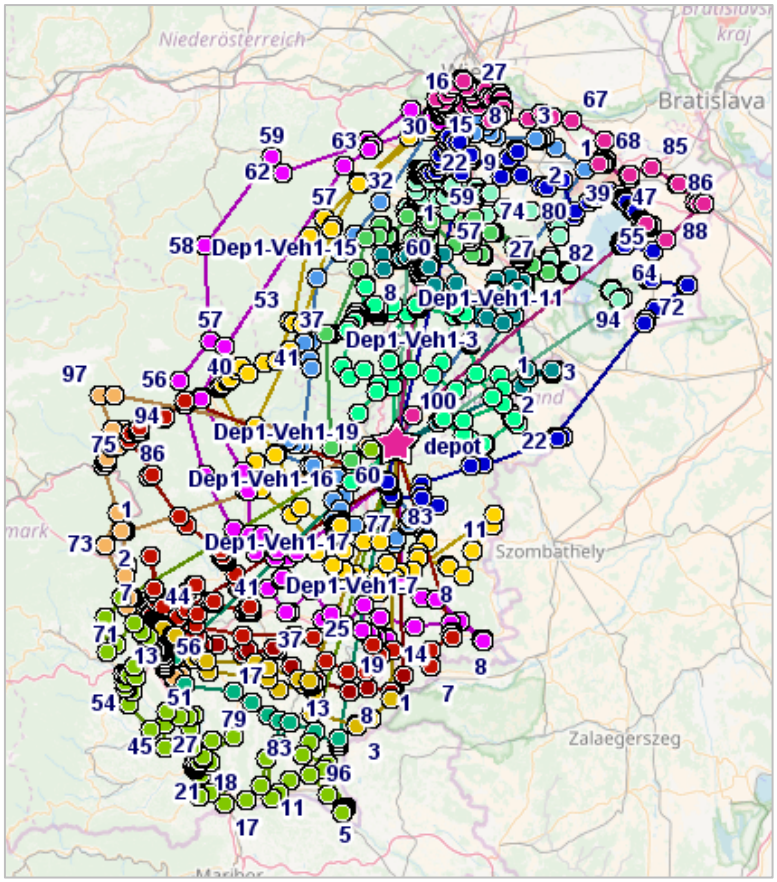

Figure 4.3. K-medoid clustering results for dataset C (Austria, k=15)

4.2 R&R results (without predefined clusters)

When there are no predefined clusters specified, *ODL Studio* generates significantly better results. Adding clustering rules to the CVRP adds extra constraints to the problem that limits the R&R method and as a result weakens the result.

The R&R results with no predefined clusters are compared to the clustering methods where the number of clusters showed the best results. The results for the three datasets are displayed below:

4.2.1 Dataset A: United Kingdom (size=100)

Travel cost			Distance travelled (km)		
K-medoid (k=3)	K-means (k=3)	R&R Method (Vehicles used = 1)	K-medoid (k=3)	K-means (k=3)	R&R Method (Vehicles used = 1)
27.43	27.91	25.05	2043	2079	1865

Table 4.4. Dataset A results of R&R method with no clusters defined compared to k-medoid and k-means

4.2.2 Dataset B: Germany (size=500)

Travel cost			Distance travelled (km)		
K-medoid (k=5)	K-means (k=5)	R&R Method (Vehicles used = 6)	K-medoid (k=5)	K-means (k=5)	R&R Method (Vehicles used = 6)
121.51	126.90	105.96	9049	9450	7892

Table 4.5. Dataset B results of R&R method with no clusters defined compared to k-medoid and k-means

4.2.3 Dataset C: Austria (size=1500)

Travel cost			Distance travelled (km)		
K-medoid (k=13)	K-means (k=13)	R&R Method (Vehicles used = 15)	K-medoid (k=13)	K-means (k=13)	R&R Method (Vehicles used = 15)
57.24	58.16	57.82	4263	4331	4306

Table 4.6. Dataset C results of R&R method with no clusters defined compared to k-medoid and k-means

4.3 Analysis of results

The interpretation of the results will be discussed as follows:

1. Processing time of methods
2. The effect of the number of clusters on results
3. Practical significance and limitations of clustering in CVRPs

4.3.1 Processing time

The processing time of both clustering methods were relatively short. The processing time for clustering never exceeded 8 seconds, even for larger datasets. The clustering algorithm allows for scalability and is robust to larger problems in terms of runtime. The routing optimisation in *ODL Studio* ranged between 15-45 seconds for dataset A and B. For dataset C, processing time increased dramatically. An attempt to model an experimental dataset of 2000 customers, could not be processed. For larger customer networks, more advanced processing capability will be required.

4.3.2 Number of clusters

The number of clusters strongly influenced the overall results. An important aspect to consider for the application to CVRPs, is the capacity or size of the vehicle. Dondo and Cerdá (2007) refer to “feasible clusters”, meaning that all the customers in one cluster should be served by a single vehicle. In the case where a particular cluster is too largely defined, an additional vehicle needs to be added to serve its customers. This adds an unnecessary constraint to the routing optimisation method, as the clusters are then subdivided.

Note that the cost to service the network, as well as the total distance travelled increases with the number of pre-defined clusters. This is because additional vehicles are added or in some cases vehicles are under-utilised. *Figure 4.4* and *Figure 4.5* illustrate how the total cost and distance increased with the number of clusters for dataset A.

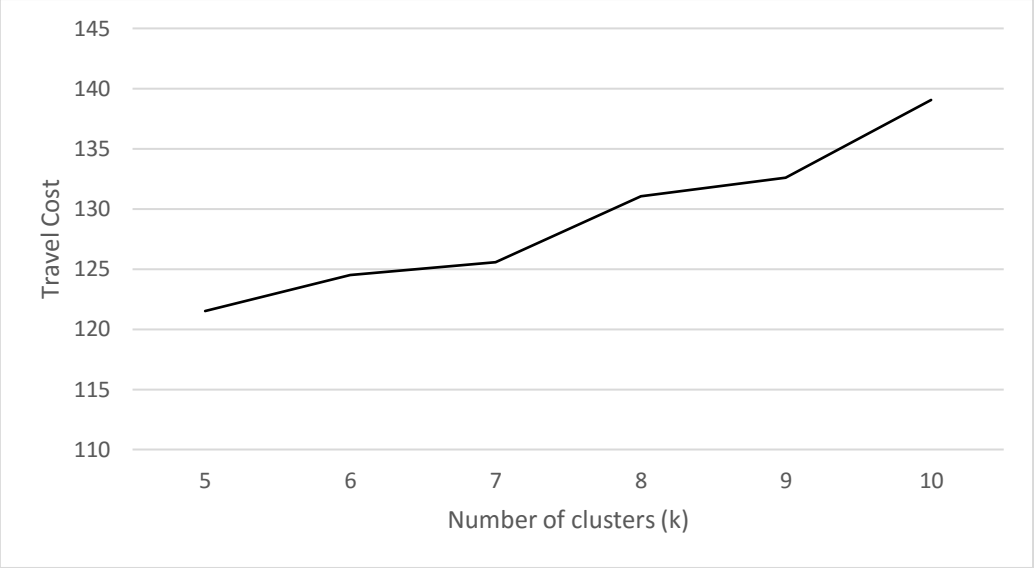

Figure 4.4. Total cost of CVRP compared to the number of clusters

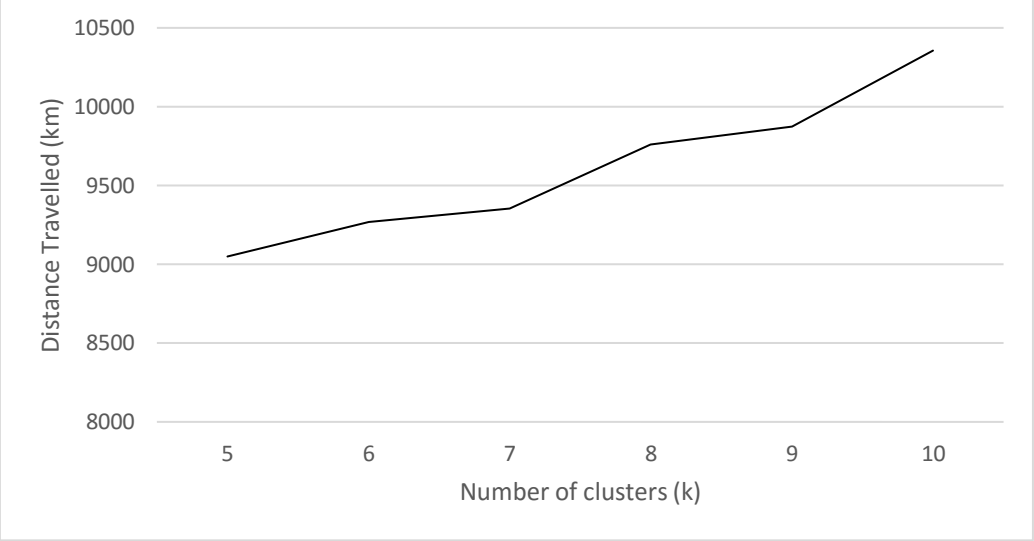

Figure 4.5. Total distance travelled compared to the number of clusters

4.2.3 Practical observations and limitations

Both clustering methods provided more logical groupings of customer locations than what the routing metaheuristic determined without clustering inputs. This is an important aspect to bear in mind, as intuitive solutions are generally preferred by customers over abstract solutions (Schrimpf *et al.*, 2000).

From an operational point of view, spatial groupings of customers is an important management tool. Clustering can aid the planning and assignment of resources to specific regions and customers. Decision-making regarding the assignment of the vehicle fleet and drivers could be directed by utilising clustering methods. The addition of new customers to the network can easily be done in a clustered network by allocating the customers to specific regions (clusters) without re-running a routing algorithm.

Clustering allows the specification of a particular number of clusters. This could be valuable in distribution problems where a number of management areas or regions need to be defined. Bührmann (2015) observed cases where clustering could be useful in determining initial solutions in Capacitated Location Routing Problems (CLRP's).

However, for vehicle routing decisions, where the sole aim is to minimise travelling cost and the distance travelled, neither k-medoid nor k-means clustering could outperform a modern metaheuristic method, such as the R&R method, as shown by *Table 4.4-4.6*. The clustering results added an extra constraint to the metaheuristic method that decreased its performance in terms of cost and distance travelled.

5. Conclusion

5.1 Conclusive remarks

In this study, the k-medoid (petal-shaped) clustering method was applied to the widely studied optimisation problem, the CVRP. In order to benchmark the results of k-medoids, a well-known clustering method, k-means, was selected. The method was applied to three datasets of different sizes to evaluate the feasibility and ideal conditions for the application thereof.

During the literature review, several route optimisation techniques were explored. To address the routing component of the study, the R&R metaheuristic method was selected due to its exceptional published results. Furthermore, the method is available through an open-source software package, *ODL Studio*. This tool was used to execute and measure the CVRP results, and allowed for the mapping of found solutions.

Overall, the data showed that k-medoids outperformed the results of k-means in the CVRP. One could conclude that k-medoid clustering could successfully be applied in instances where logical clustered solutions are preferred over pure metaheuristic methods.

However, in context of day-to-day vehicle routing decisions, the use of clustering in conjunction with modern metaheuristics seems situational when the main objective is merely to find minimum cost solutions. Advanced metaheuristics, such as the R&R method used in this study, offered better optimisation solutions in terms of cost and distance when used without predefined groupings assigned. Adjustments could possibly be made to metaheuristic routing solutions to make it more intuitive and logical to implement.

The results suggested that the application of k-medoid, and possibly other clustering methods, should be limited to high-level distribution network planning and the assignment of resources. During the modelling of the studied methods, it was clear that clustering enables effective scalability in terms of processing times and could be advantageous for large datasets.

5.2 Recommendations for future research

The following areas of research were identified for future work:

- The R&R principle presented exceptional results and can be tested on other optimisation problems. VRPs with Time-Windows (VRPTW) could be a useful problem to solve using the R&R method.
- The determination of the optimal number of clusters should be explored further. Methods or techniques that are specifically relevant for distribution networks should be investigated.
- Given the strain of large datasets on processing resources, the impact of clustering for distribution network problems for massive datasets should be explored.
- The effect of clustering on the initial design of distribution networks e.g. determination of vehicle fleet size, geographical location of depot, and the assignment of specific resources to specific regions should be evaluated.

6. References

Blum, C. (2005) ‘Ant colony optimization: Introduction and recent trends’, *Physics of Life Reviews*, pp. 353–373.

Blum, C. and Roli, A. (2003) ‘Metaheuristics in combinatorial optimization: overview and conceptual comparison’, *ACM Computing Surveys*, 35(3), pp. 189–213.

Bührmann, J. (2015) *The effects of clustering on the Capacitated Location-Routing Problem*. PhD. University of the Witwatersrand.

Caccetta, L., Alameen, M. and Abdul-Niby, M. (2013) ‘An Improved Clarke and Wright Algorithm to Solve the Capacitated Vehicle Routing Problem’, *Technology & Applied Science Research*, 3(2), pp. 413–415.

Christofides, N. and Eilon, S. (1969) ‘An algorithm for the vehicle dispatching problems’, *Operational Research Quarterly*, 20(3), pp. 309–318.

Dantzig, G. B. and Ramser, J. H. (1959) ‘The Truck Dispatching Problem’, *Management Science*, 6(1), pp. 80–91.

Dondo, R. and Cerdá, J. (2007) ‘A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows’, *European Journal of Operational Research*.

Dorigo, M., Maniezzo, V. and Colorni, A. (1996) ‘Ant system: Optimization by a colony of cooperating agents’, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 26(1), pp. 29–41.

Everitt, B. S., Landau, S., Leese, M. and Stahl, D. (2011) *Cluster Analysis, Quality and Quantity*.

Fisher, M. L. and Jaikumar, R. (1981) ‘A generalized assignment heuristic for vehicle routing’, *Networks*, 11(2), pp. 109–124.

Glover, F. (1989) ‘Tabu Search - Part I’, *ORSA journal on Computing*, 2 1(3), pp. 4–32. Gore, P. A. (2000) ‘Cluster Analysis’, *Handbook of Applied Multivariate Statistics and Mathematical Modeling*, (323), pp. 297–321.

Henderson, D., Jacobson, S. H. and Johnson, A. W. (2003) ‘The theory and practice of simulated annealing’, *Handbook of metaheuristics*, pp. 287–319.

Hilier, F. and Lieberman, G. (2015) ‘Introduction to Operational Research’, in *Introduction to Operational Research*, pp. 731–784.

Jain, A. K. (2010) ‘Data clustering: 50 years beyond K-means’, *Pattern Recognition Letters*, 31(8), pp. 651–666.

Kaufman, L. and Rousseeuw, P. J. (1990) ‘Partitioning Around Medoids (Program PAM)’, *Finding Groups in Data: An Introduction to Clustering Analysis*, pp. 68–125.

Kohavi, R. (1995) ‘A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection’, in *Appears in the International Joint Conference on Articial Intelligence (IJCAI)*, pp. 1–7.

Land, A. H. and Doig, A. G. (1960) ‘An automatic method for solving discrete programming problems’, *Econometrica*, 28(3), pp. 497–520.

Laporte, G. (1992) ‘The Vehicle Routing Problem: An overview of exact and approximate algorithms’, *European Journal of Operational Research*, 59(2), pp. 345–358.

Laporte, G. (2009) ‘Fifty Years of Vehicle Routing’, *Transportation Science*, 43(4), pp. 408–416.

Laporte, G., Gendreau, M., Potvin, J.-Y. and Semet, F. (2000) ‘Classical and modern heuristics for the vehicle routing problem’, *International Transactions in Operational Research*, 7(4–5), pp. 285–300.

Miller, L. R. and Gillett, B. E. (1974) ‘A Heuristic Algorithm for the Vehicle-Dispatch Problem’, *Operations Research*, pp. 340–349.

Mohammed, M. A., Ahmad, M. S. and Mostafa, S. a. (2012) ‘Using Genetic Algorithm in implementing Capacitated Vehicle Routing Problem’, *2012 International Conference on Computer & Information Science (ICCIS)*, pp. 257–262.

Moolman, A., Koen, K. and v.d. Westhuizen, J. (2010) ‘Activity-based costing for vehicle routing problems’, *SAJIE*, 21(2), pp. 161–171.

ODL Studio (2014) *Open Door Logistics - Intelligent software for vehicle routing*. Available at: http://www.opendoorlogistics.com/software/odl-studio/ (Accessed: 7 November 2017).

OSGB (2010) ‘A guide to coordinate systems in Great Britain’, p. 43. Available at: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:A+guide+to+coordinate+systems+in+Great+Britain#7.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, É. (2012) ‘Scikit-learn: Machine Learning in Python’, *Journal of Machine Learning Research*, 12, pp. 2825–2830.

Python Software Foundation (2001) *Python.org*. Available at: https://www.python.org/ (Accessed: 7 November 2017).

Ryan, D. M., Hjorring, C. and Glover, F. (1993) 'Extensions of the Petal Method for Vehicle Routeing', *The Journal of the Operational Research Society*, 44(3), pp. 289–296.

Salvador, S. and Chan, P. (2003) 'Determining the Number of Clusters/Segments in Hierarchical Clustering / Segmentation Algorithms', *Work*, p. 20.

Schrimpf, G., Schneider, J., Stamm-Wilbrandt, H. and Dueck, G. (2000) 'Record Breaking Optimization Results Using the Ruin and Recreate Principle', *Journal of Computational Physics*, 159(2), pp. 139–171.

Wang, Y., Ma, X., Xu, M., Wang, Y. and Liu, Y. (2015) 'Vehicle routing problem based on a fuzzy customer clustering approach for logistics network optimization', *Journal of Intelligent & Fuzzy Systems*, 29(4), pp. 1427–1442.

Appendix A

Python k-medoid method code

The k-medoid code was developed based on methods from a combination of sources, such as Kaufman and Rousseeuw (1990). However, it had to be adjusted extensively for the purpose of this study. The code below includes functions to read and write data to Excel, calculation of the k-medoid cluster indices, and the plotting of the calculated clusters.

```
from sklearn.metrics.pairwise import pairwise_distances
import numpy as np
import random
import kmedoids
import ReadData
import WriteData
import pylab as pl
import matplotlib.cm as cm

#Points in dataset
data = ReadData.ReadData()

#Distance matrix
D = pairwise_distances(data, metric='euclidean')

# Run kMedoids function (D, k, tmax=100):
M, C = kmedoids.kMedoids(D,7)

print('medoids:')

#Plot data
#Create array of x values
x = data[:,0]
print(x)
#Create array of y values
y = data[:,1]
print(y)
#pl.plot(x, y,'r*')
```

```
# set axes
#plt.axis([0, 6, 0, 20])
# show the plot on the screen
for point_idx in M:
    print( data[point_idx] )
print('')
print('clustering result:')

# Define colour array for each k
col                                                                    =
['r*','b*','g*','m*','k*','c*','y*','k*','b*','r*','r*','b*','g*','m*','k*
','c*','y*','k*','b*','r*']

for label in C:
    for point_idx in C[label]:
        print('label {0}:  {1}'.format(label, data[point_idx]))
        pl.plot(x[point_idx], y[point_idx],col[label])

WriteData.WriteData(C,x,y)

-----------------------------------------------------------------
def ReadData():
    from openpyxl import load_workbook
    import numpy as np

    #Read from excel file, specific sheet
    wb = load_workbook('test.xlsx')
    sheet_1 = wb.get_sheet_by_name('Sheet1')
    a = sheet_1.max_row

    #Define arrays
    D = np.zeros((a,2))

    #Read columns into arrays
    for i in range(0,a):
        D[i,0]=sheet_1.cell(row=i+1, column=1).value
        D[i,1]=sheet_1.cell(row=i+1, column=2).value

   #Return results
    return D
```

```
---------------------------------------------------------------------

import numpy as np
import random

def kMedoids(D, k, tmax=100):
    #Calculate size of distance matrix D
    m, n = D.shape

    if k > n:
        raise Exception('too many medoids')
    #Randomly initialize array of indices
    M = np.arange(n)
    np.random.shuffle(M)
    M = np.sort(M[:k])

    #Duplicate medoid indices array
    Mnew = np.copy(M)
    #Initialize set to represent clusters
    C = {}
    for t in range(tmax):
        #Determine clusters
        J = np.argmin(D[:,M], axis=1)
        for kappa in range(k):
            C[kappa] = np.where(J==kappa)[0]
        #Update cluster medoids
        for kappa in range(k):
            J = np.mean(D[np.ix_(C[kappa],C[kappa])],axis=1)
            j = np.argmin(J)
            Mnew[kappa] = C[kappa][j]
        np.sort(Mnew)
        #Check for convergence
        if np.array_equal(M, Mnew):
            break
        M = np.copy(Mnew)
    else:
        #Final update of cluster allocations
        J = np.argmin(D[:,M], axis=1)
        for kappa in range(k):
```

```
            C[kappa] = np.where(J==kappa)[0]

    #Return results
    return M, C

---------------------------------------------------------------------

def WriteData(C,x,y):
    from xlwt import Workbook
    import numpy as np

    #Read  from excel file, specific sheet
    wb = Workbook()
    ws = wb.add_sheet('ClusteringResults')
    wb.save('Clustering results test.xls')

    a = len(x)
    print("a= ",a)

    row = 0

    for label in C:
        for point_idx in C[label]:
            ws.write(row,1,x[point_idx])
            ws.write(row,2,y[point_idx])
            ws.write(row,3,label)
            ws.write(row,0,str(point_idx))
            row = row+1

     wb.save('Clustering results test.xls')

---------------------------------------------------------------------
```

Appendix B

Python k-means method code

The Scikit-learn Machine Learning library (Pedregosa *et al.*, 2012) includes the k-means clustering algorithm. The module was imported in Python. In the code below, the same functions as in *Appendix A* are used to read and write data, and the k-means function is called from the Scikit-learn module.

```
from sklearn.cluster import KMeans
import numpy as np
import ReadData
import WriteData

data = ReadData.ReadData()
#D = pairwise_distances(data, metric='euclidean')

print('data',data)
kmeans = KMeans(n_clusters=X, random_state=0).fit(data)
L = kmeans.labels_
print(L)

WriteData.WriteData(data,L)
```

Appendix C

CVRP input tables

Samples of the CVRP input tables in ODL Studio are shown below. *Table 7.1* contains the customer locations, with coordinates and demand quantities. *Table 7.2* describes the input parameters of the vehicle fleet.

	id	latitude	longitude	quantity
1	Stop1	51.386	12.45	50
2	Stop2	52.504	13.319	50
3	Stop3	53.563	9.916	50
4	Stop4	51.305	11.986	50
5	Stop5	51.937	11.305	50
6	Stop6	51.909	8.677	50
7	Stop7	52.372	9.741	50
8	Stop8	53.547	9.988	50
9	Stop9	53.553	9.987	50
10	Stop10	54.134	8.857	50
11	Stop11	52.537	13.619	50
12	Stop12	53.558	9.989	50
13	Stop13	50.345	8.898	50
14	Stop14	53.728	9.911	50
15	Stop15	52.462	13.315	50
16	Stop16	52.318	12.744	50
17	Stop17	52.569	13.497	50
18	Stop18	53.072	8.829	50
19	Stop19	51.642	9.56	50
20	Stop20	54.045	9.481	50

Table 7.1. Sample of customer locations input table in ODL Studio

	vehicle-name	vehicle-id	start-latitude	start-longitude	end-latitude	end-longitude	capacity	cost-per-km	fixed-cost
1	Dep1-Veh0	Dep1-Veh0	51.714	10.353	51.714	10.353	500	0.001	100
2	Dep1-Veh1	Dep1-Veh1	51.714	10.353	51.714	10.353	500	0.001	100
3	Dep1-Veh2	Dep1-Veh2	51.714	10.353	51.714	10.353	500	0.001	100
4	Dep1-Veh3	Dep1-Veh3	51.714	10.353	51.714	10.353	500	0.001	100
5	Dep1-Veh4	Dep1-Veh4	51.714	10.353	51.714	10.353	500	0.001	100
6	Dep1-Veh5	Dep1-Veh5	51.714	10.353	51.714	10.353	500	0.001	100
7	Dep1-Veh6	Dep1-Veh6	51.714	10.353	51.714	10.353	500	0.001	100
8	Dep1-Veh7	Dep1-Veh7	51.714	10.353	51.714	10.353	500	0.001	100
9	Dep1-Veh8	Dep1-Veh8	51.714	10.353	51.714	10.353	500	0.001	100
10	Dep1-Veh9	Dep1-Veh9	51.714	10.353	51.714	10.353	500	0.001	100
11	Dep1-Veh10	Dep1-Veh10	51.714	10.353	51.714	10.353	500	0.001	100
12	Dep1-Veh11	Dep1-Veh11	51.714	10.353	51.714	10.353	500	0.001	100

Table 7.2. Sample of vehicle fleet input table in ODL Studio