

ROUTE OPTIMISATION FOR THE SOUTH AFRICAN POST OFFICE SoC LTD

by

ALEMAYEHU TAFESSE

Student number : 11031850

Submitted in partial fulfilment of the requirements for the degree of

BACHELORS OF INDUSTRIAL AND SYSTEMS ENGINEERING

in the

FACULTY OF ENGINEERING, BUILT ENVIRONMENT AND INFORMATION TECHNOLOGY

UNIVERSITY OF PRETORIA

SOUTH AFRICA

October 2015

Abstract

Post Offices globally are suffering from a decrease in revenues as a result of a decreasing customer base and an increasing delivery network. This is because of their obligation to deliver mail at the same price irrespective of location, usually mandated by governments. Therefore Post Offices need to look at making each aspect of their core business more efficient and effective. This project focuses on the methods used to route mail from Post Office depots to delivery points (such as houses).

An investigation was completed on the current methods used by the South African Post Office to plan postmen routes. The method, which updated routes once every 2 to 3 years, was deemed inefficient as it lacked any flexibility and couldn't account for everyday eventualities such as postman absenteeism and vehicle breakdown.

A literature review presented herein identified arc routing as the most relevant branch of operations research for this project. Specifically it was found that modelling the problem as a Capacitated Arc Routing problem that was solved using a path scanning heuristic would make the best use of the students skill set, the time available and would approximate a good solution for the Post Office.

This report describes the chosen solution method, the tests conducted to ensure its effectiveness and the final solutions generated. Finally a brief chapter on possible future work is given.

Contents

1	Introduction	1
1.1	The Post Office	1
1.1.1	Global Trends	1
1.1.2	Mail delivery	2
1.1.3	Route Planning	2
1.2	Problem Definition	2
1.3	The Pilot Depot	2
1.3.1	The chosen pilot depot	3
1.4	Research Design	5
1.5	Research Methodology	6
1.5.1	Data Analysis and research	6
1.5.2	System formulation	6
1.5.3	Pilot location	6
1.5.4	Model evaluation and presentation of solution	6
1.6	Document Structure	6
2	Literature Review	7
2.1	Chinese Postman Problems	8
2.1.1	Relevance to the Post Office	8
2.2	Rural Postman Problems	8
2.2.1	Relevance to the Post Office	9
2.3	Solutions to Rural postman problems	9
2.3.1	Directed, undirected and Mixed RPPs	9
2.3.2	Solutions to windy Postman Problems	11
2.3.3	Concerns and Conclusion	12
2.4	The Capacitated arc routing problem	12
2.4.1	Relevance to the Post Office	13
2.5	Solution Methods for the CARP	13
2.5.1	Performance of the heuristics	15
2.5.2	Conclusion	16
2.6	Alternative solutions	16
2.6.1	Conclusion	17
2.7	Conclusion	18

3	Solution Design	19
3.1	The chosen method	19
3.2	Representing the problem on a graph	19
3.3	Determining the capacity and demand	20
3.4	Interpreting the solution	21
3.5	Extended path scanning	21
3.5.1	Rules 1 and 2	21
3.5.2	Rules 3 and 4	21
3.5.3	Rule 5	22
3.6	Concerns with the tie break rules	22
3.7	Path scanning operators	23
3.8	Building the route	24
3.8.1	Testing the routes	25
4	Testing the chosen method	26
4.1	Rational for testing	26
4.2	The test area	26
4.3	Test 1: Effect of the tie break rules	26
4.3.1	Scenario 1: Demand as an average	27
4.3.2	Scenario 2 : Demand equal to service time	28
4.3.3	Scenario 3: Demand equal to service time using Rule 5 only	28
4.3.4	Results and considerations	28
4.4	Test 2 : Modelling with only arcs	31
4.4.1	Results and Considerations	32
4.5	Test 3: Off the shelf software	32
4.5.1	Results and considerations	34
4.6	Conclusion	34
4.7	Additional Considerations: Similarity of routes	34
4.8	Conclusion	35
5	Generating final routes	36
5.1	Final Routes	36
5.1.1	Analysing the Routes	36
5.1.2	Effect of a change in capacity	37
5.1.3	Change in model size affecting time to run	38
5.2	Conclusion	41
6	Future work	42
6.1	Future work with CARP	42
6.2	Conclusion	43

List of Figures

1.1	Steps to implement pilot	3
1.2	The merge walk plan	4
1.3	An example of the shelves at the depot	4
1.4	An example of a Post Office time sheet	5
2.1	Initial graph with both required and non-required edges	10
2.2	Transformed graph with required edges and even degree nodes	11
4.1	Total area that was modelled	27
4.2	Representation of the routes	27
4.3	The results of scenario 1	28
4.4	The results of scenario 2. Note some deadheading is omitted for graph simplicity. Instead some sections connected by deadheading have been represented by a square on the nodes that are meant to be linked.	29
4.5	The results of scenario 3.	29
4.6	Comparison of the overall efficiency of scenarios with an increase in problem size .	30
4.7	Effect of dead heading on a serviced edge. Replacing the edges with arcs would result in the efficiency of the route increasing from 50% to 100%	31
4.8	RouteXL generated route	32
4.9	Route Planner generated route	33
4.10	Model generated route	33
4.11	Result of several tests	34
5.1	Route 20, the least efficient route generated through model. As in Figure 4.4 the connect deadhead blocks are used. The two instances where they are used are in two separate colours for simplicity.	39
5.2	Route 15, the most efficient route generated through model	40

List of Tables

4.1	Results from the first test	30
4.2	Results from the first test with all deadheaded required edges replaced by arcs. This was done by analysing the maps drawn, removing all deadheading on a serviced edge and replacing it with two arcs.	31
4.3	Results from the third test	34
5.1	Breakdown of all routes generated	36
5.2	Full list of routes generated	37
5.3	Breakdown of change in capacity test	38
5.4	Time taken to develop routes	38

List of Algorithms

1	Path scanning heuristic	15
2	Complete path scanning heuristic	19
3	Rule 1	21
4	Rule 3	22
5	Rule 5	22
6	Test load	23
7	Find Nearest Arc	23
8	Check Nearest Arc	24
9	Add Arc	24
10	Build route	25

Chapter 1

Introduction

1.1 The Post Office

The Post Office is South Africa's national mail carrier and is mandated by law to deliver mail within the country irrespective of location. To achieve this, massive infrastructure investments are required in order to keep up with the urban sprawl indicative of a developing nation and to reach communities previously neglected. These investments will total R3.5 billion over the course of 2014, 2015 and 2016 combined. During 2014, 50 postal outlets were opened across the country taking the number of Post Office branches and depots in operation to over 3000. These outlets service just under 11 million addresses across the country of which 1.2 million new addresses were added in 2014. The effective planning of delivery routes for mail has proven to be problematic, partly because of the rate at which new addresses are added to the delivery network and also because of the static planning methods used.

1.1.1 Global Trends

The problem of diminishing revenue is not unique to the South African Post Office. The United States Postal Service (USPS), arguably the largest postal service in the world, has also identified diminishing consumer bases and growing delivery networks due to its universal service obligation. USPS recently attempted to experiment with a dynamic routing system (Waxer, 2014) for their parcel delivery and found that huge time savings were made since postmen did not have to plan out their delivery routes daily (Guerrero, 2015). Direct competitors such as UPS and Fedex have managed to be profitable due to the implementation of dynamic routing and the integration of their routing systems with real time GPS feedback. Implementing solutions such as this is expensive, and Post Offices worldwide currently do not have the money available to implement these systems on a large scale. Cost effective, and efficient alternatives need to be found to remain competitive.

Before any improvements can be suggested, pause pause must be taken to analyse how the South African Post Office currently operates.

1.1.2 Mail delivery

Mail is delivered to a depot and is sorted according to a postman's route. The postman then either begins his delivery tour from the depot, or is dropped off to the starting point of his tour with a car. Currently postmen either walk, use a motorbike or use a bicycle to deliver mail. A postman's tour is planned by performance inspectors, who take a variety of factors unique to each depot into consideration when planning delivery routes. Therefore each route is unique to a Post Office depot.

1.1.3 Route Planning

Routes are planned taking various constraints into account. For new outlets planners first gather information about the immediate surroundings of the post office, some of the constraints include the terrain, delivery points, capacity of the postman, distance and transport method. Thereafter a delivery route is planned and remains unchanged for approximately 2-3 years until its next update. The gap between route updates has proven to be problematic as there are a variety of constraints that are dynamic in nature. For example the route takes the number of postman available to be static which is problematic if routes need to be modified due to a worker strike, the retirement of a postman, the purchasing or loss of bicycles and any new roads built/added to delivery network etc.

1.2 Problem Definition

The current methods used to route mail for the Post Office were deemed inefficient due to the fact that they were static in nature, left a large paper trail and were heavily dependant upon work done by people outside individual depots. A dynamic routing system was needed that took as an input the delivery network and its associated costs or distances and delivered as an output an optimized route.

Due to the size of the Post Office it would have been infeasible to test the validity of the system across all its outlets, therefore the system was tested at a pilot depot.

1.3 The Pilot Depot

Using a pilot depot to test the proposed system allowed the student the opportunity to make corrections where necessary, validate and demonstrate the system to the Post Office.

A nine step plan was used to implement the proposed solution at the depot. As seen from Figure 1.1, the first step in the process was creating the pilot plan. The pilot plan, which took the form of the project proposal and preliminary report, described the chosen solution method and showed how the system would be implemented. This allowed the student the opportunity to ensure support and gave the Post Office the opportunity to identify an appropriate pilot depot, in steps two, three and four respectively. Relevant information about the depot such as the streets serviced, distances and the number of postmen working was collected in the fifth. After running the model the routes generated were compared to the routes currently in use and the results analysed. The final results were then presented to the Post Office.

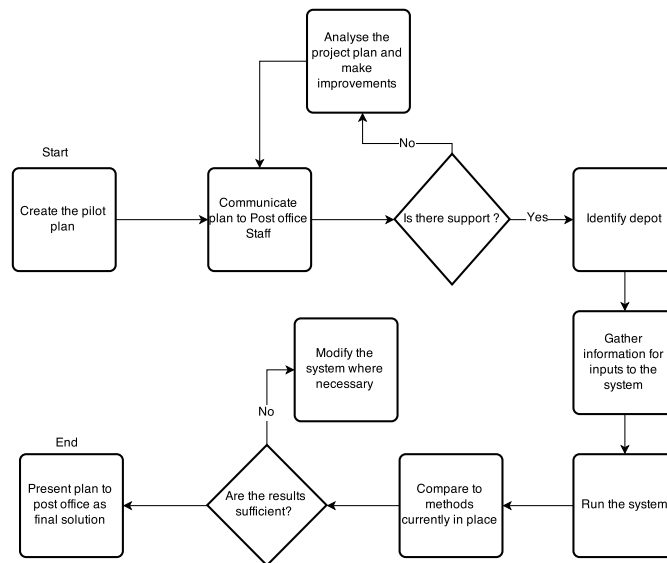


Figure 1.1: Steps to implement pilot

1.3.1 The chosen pilot depot

The depot that was chosen was the Waterkloof depot located in Milner street, Pretoria East. The depot services approximately 800 streets subdivided into 35 routes of which 1 is done on foot, 2 with a motorbike and the remainder by bicycles. There is however a shortage of postmen, therefore the excess routes are usually split up between the remaining postmen creating merged walk e.g. If route 33 does not have a postman to service it, it is cut in half and shared between the postmen who service routes 32 and 34. This is a clear example of a situation where a flexible and dynamic routing system is required. There is no technology available at the depot to make the changes to the routes, an A4 sheet with a brief description as shown in Figure 1.2 was the only instruction visible at the depot.

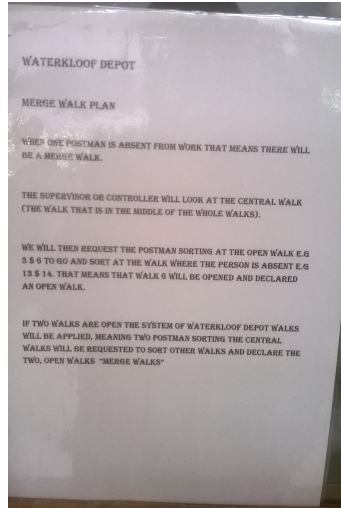


Figure 1.2: The merge walk plan

Routes are represented in the Post Office depot by a series of shelves that have the first delivery point on the top left corner and the last delivery point on the bottom right. In the morning a postman takes mail that was sent from the distribution centre and arranges it on these shelves in order to prepare for his delivery route. There is a limit to the mass of mail a postman can carry at any given time. If the mail for a walk is heavier than the limit, the excess mail is placed in a separate bag, called a time bag, and dropped off at a predetermined location along the postman's route in order to be picked up later.



Figure 1.3: An example of the shelves at the depot

It was necessary to determine the total time spent by the postmen delivering mail during an 8 hour day. Through the analysis of time sheets, an example of which is available in Figure 1.4, it was found that postmen spend on average an hour in the mornings sorting mail and preparing for their walks, with another hour taken up on miscellaneous activities such as using the bathroom or taking a drink break. Therefore all models generated in this report have been built assuming a 6 hour delivery route.

Figure 1.4: An example of a Post Office time sheet

1.4 Research Design

The main aim of this project was to identify and analyse a real world problem and through the use of mathematical models improve or redesign the solution method in use. A model was implemented that used a heuristic to approximate optimal routes for the postmen of the South African Post Office.

Due to the nature of the problem and the solution method chosen, this project could be defined as an operations research project. In the ‘Encyclopaedia of operations research and management science’, Gass and Harris (2001) define operations research as “The application of methods of science to complex problems arising in the direction and management of large systems of men, machine, materials and money in industry, business, government and defence”. The institute of operations research and management sciences simplifies this definition to “A discipline that deals with the application of advanced analytical methods to help make better decisions” (INFORMS, 2015).

The model took as an input a list of required streets for service, as well as the time to service the streets, the demand of the streets and the cost to travel along the streets. A heuristic was then used to approximate a solution before the optimal routes were given as a result.

1.5 Research Methodology

In order to ensure the successful completion of the project, a 4 step plan was created. The plan was built from the pilot plan and from the steps for the successful completion of an operations research project by Tiwari and Shandilya (2006). Broadly defined, the plan began with a detailed study of arc routing problems, the formulation and testing of an adequate solution and implementation of the solution method chosen.

1.5.1 Data Analysis and research

Research was conducted in order to fully understand the different methods available to solve routing problems. Methods identified were evaluated against the current skill set of the student, the time available for the project and the possibility for further improvement in future.

1.5.2 System formulation

Taking the data available from the Post Office and the solutions available in literature, it had to be decided if a new algorithm needed to be formulated, or an existing algorithm/solution need was to be used, with the latter identified as the most feasible.

1.5.3 Pilot location

The Waterkloof Post Office depot was chosen to serve as a pilot location for the project. The pilot depot was evaluated in order to identify all relevant information for the project. This included but was not limited to:

- The total number of streets the Post Office services.
- The number of streets in the Post Offices deliver network.
- Current methods in place to decide routes.
- Current routes in use.

1.5.4 Model evaluation and presentation of solution

Finally the routes were formulated, tested and evaluated against several alternative solution methods (i.e. off the shelf software) in order to validate its effectiveness.

1.6 Document Structure

A literature review that introduces the problem and solution methods available is presented in Chapter 2. The different methods are evaluated against one another and a final method chosen. This method is presented in full in Chapter 3. Chapter 4 evaluates the effectiveness of the solution method with the final results presented in Chapter 5. Chapter 6 states any possible future work for the project.

Chapter 2

Literature Review

Assume a postman has to plan his delivery route that starts at his depot and that this route must include select streets while other streets are optional. Furthermore assume that there is a cost associated with travel between the houses. Before starting the journey, the postman has some questions that he needs to answer. What would be the cheapest route that he can use that includes all the houses he has to deliver mail to? Would this route also be the shortest route? Would he need to use a road more than once?

This type of problem is modelled as an arc routing problem, where a problem such as the hypothetical one proposed will be transformed into a graph, and the optimal route found. This report investigated three different types of arc routing problems, Rural postman problems, Chinese postman problems and Capacitated arc routing problems.

Routing problems can be traced back to the seven bridges of Königsberg problem of 1736. Mathematician Leonard Euler attempted to find a path that visited the seven bridges of Königsberg and that started and ended at the same point (known as a eulerian tour/path). Although he did not find such a path for the problem, he did find the conditions necessary to find a solution to a routing problem. Eulers theorem states that a graph G is eulerian if and only if every node has an even degree. To model an arc routing problem such as the one faced by Euler or the postman in our example, a graph G as defined in Eiselt et al. (1995a) is introduced. $G = (V, E)$ represents the graph with the vertex set $V = \{v_1, v_2, \dots, v_n\}$, representing the street intersections connecting roads, and the edge set $E = \{(v_i, v_j) : v_i, v_j \in V \text{ and } i \neq j\}$, representing the roads in our example. Finally C_{ij} is represented as the cost of traversing an edge (v_i, v_j) .

Modelling the graph with edges allows for two way travel per edge (similar to a two way street) and the graph is said to be undirected. If the problem is defined to allow for only one-way travel it is modelled with arcs instead of edges and the graph is said to be directed. Situations may arise where the postman in our example would only have to service some of the streets in his delivery network, to model this, a set E_R is defined as a subset of E and contains all the required edges. The same procedure is followed with arcs with A_R defined as a subset of A . If $E_R \neq \emptyset$ and $E_R \neq E$ (likewise for arcs), meaning that there are some edges and arcs that need to be serviced while there are some that don't, the problem is modelled as a rural postman problem. Most arc routing applications are of the RPP type (Eiselt et al., 1995b). If $A_R = A$ or $E_R = E$ meaning all the arcs or edges need to be traversed, the problem is modelled as a Chinese postman problem.

2.1 Chinese Postman Problems

One of the best known applications of Arc Routing is the Chinese postman problem (CPP). For the graph $G = (V, E)$ comprising of a vertex set V and an edge set E , the Chinese postman problem can be described as having to find a tour that visits each edge on at least once. Using the hypothetical postman described earlier, CPPs arise when the objective of the postman is to find an optimal tour that allows him to visit every street in his delivery network and return to his depot. An optimal Chinese postman tour is a tour of minimal cost (Farahani, 2012). CPPs come in a range of categories, each addressing a specific problem that may arise in industry.

Directed A Directed CPP is comprised entirely of arcs (Thimbleby, 2003) . In the postman example given above, this would mean that the postman's route is comprised entirely of one way roads.

Undirected As explained by Eiselt et al. (1995a) in an Undirected CPP all the edges in the graph have no set direction (i.e. the graph is comprised entirely of edges). In the postman example, this would mean the postman's route is comprised entirely of 2 way roads.

Hierarchical The Hierarchical CPP (Ghani and Improt, 2000) arises when there needs to be a precedent set over the arcs that need to be toured.

Time constrained The Time constrained CPP (Wang and Wen, 2002) can be used if the Post Office wants to impose a time limit within which the postman must service the houses.

Mixed As stated in Minioka (1979) the Mixed CPP is comprised of both arcs and edges. This type of CPP is relevant to the South African Post Office as situations arise where the motorbikes need to travel along 1 way and 2 way roads.

2.1.1 Relevance to the Post Office

The time constrained CPP on a mixed graph would have the most relevance to the Post Office. Looking back at the original requirements of the Post Office, it was made clear that they wanted a routing system that ensured that the postmen were given routes that made certain they worked the required 8 hours a day.

However it was decided that the CPP and its variants were not flexible enough to solve the Post Offices routing needs. This is because situations do arise where the Post Office does not need to service every street in its delivery network or wishes to avoid certain highways/freeways that fall within its delivery network. The RPP was identified as a good alternative and investigated.

2.2 Rural Postman Problems

For the graph $G = (V, E)$ comprising of a vertex set V an edge set E and a set of required edges $E_R \subset E$, the rural postman problem can be described as having to find a tour that visits each required edge at least once. Eiselt et al. (1995b) describes the 3 broad categories of RPP that will be looked at in this chapter:

Undirected The undirected RPP and undirected CPP are similar in that all the edges in the graph have no set direction.

Mixed This is a graph comprised of both arcs and edges, where the optimal tour will only need to visit a set or required edges and arcs.

Windy Each of the arcs in a Windy rural postman problem (Benavent et al., 2007), WRPP, has a cost associated to it according to the direction of travel. Therefore in order to discourage movement in a specific direction, the cost to travel in that direction can be increased significantly (acting like a gust of wind opposite to the direction of motion). There are variations to WRPPs, including WRPPs with zigzag service (Irnich, 2008) and WRPPs that incorporate multiple postman into their solution called Min Max K postman WRPP, as studied by Benavent et al. (2010) and Benavent et al. (2009) which will be discussed in this paper.

2.2.1 Relevance to the Post Office

Windy RPPs were singled out as having the most relevant application to the Post Office. Not only does this problem type allow for the modelling of fatigue, inclines in roads and costs (as is currently done in the Post Office), it also allowed for the signalling out of required streets which was an advantage over the CPP.

After having been identified as relevant, research was completed on the RPP and its solution methods, and is presented below.

2.3 Solutions to Rural postman problems

RPPs are classified as NP hard problems (Lenstra and Kan, 1981), meaning that while exact solutions do exist for RPP problems the time taken to find them increases exponentially as the size of the problem increases. However approximate solutions can be found by using heuristics and metaheuristics. The solution methods of RPPs needed to be investigated in order to determine the relevance of different solutions to the Post Offices needs.

The methods used to solve general directed, undirected and mixed RPPs are discussed in this chapter as they form the basis of most arc routing solutions. Thereafter the solution methods of WRPPs are looked at, these solution methods are relevant to the project as they allow us to model factors that affect the postman, such the gradient of a street or fatigue, by increasing the cost component of the edges of the graph in the direction of the planners choice.

2.3.1 Directed, undirected and Mixed RPPs

Generally solutions to RPPs start with a transformation of some graph $G = (V, E)$, into a graph $G_R = (V_R, E_R)$ comprised of only the required edges, E_R , and vertices, V_R . Thereafter the optimal path that visits each required edge at least once is found. If all the edges are required edges, that is if $E = E_R$, the problem can be modelled as a CPP. One of the most well-known approaches to solving problems of this nature is given by Edmonds and Johnson (1973) and involves transforming the graph $G = (V, E)$ into a graph of even degree nodes. If the graph has odd degree nodes, they can be converted by applying the minimum-matching algorithm. This is first done by identifying all the odd degree vertices in the graph, and calculating the shortest distance between them. This can be done by sight; or by using algorithms such as Dijkstra's algorithm (Dijkstra, 1959) . After

the shortest distance is found, the pairs are connected by constructing additional edges to the graph. From this graph the shortest path that visits each vertex and required edge found.

The heuristic proposed by Frederickson et al. (1976) can solve directed and undirected RPPs using a three step method.

1. Ensure that the graph is a connected graph. If the graph is not connected, find the minimum spanning tree to connect components using methods such as Krusals algorithm.
2. Ensure the graph is eulerian. If the graph is not eulerian use the minimum matching algorithm to create a eulerian graph.
3. Find an optimal tour.

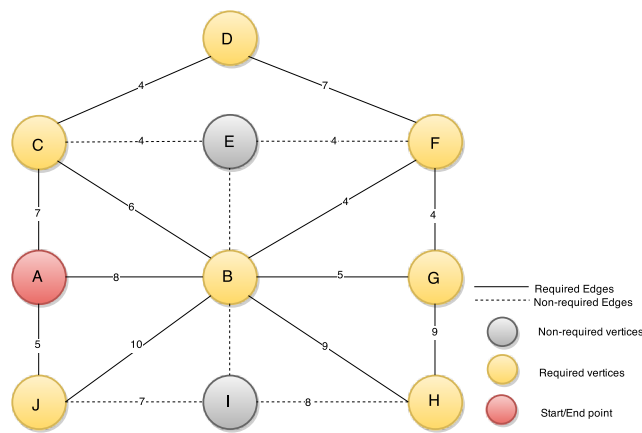


Figure 2.1: Initial graph with both required and non-required edges

Knowing how to transform a graph and find the solution on the transformed graph forms the basis of most solution methods to RPPs.

Corbera et al. (2000) solves the MRPP by using both an algorithm and a tabu search heuristic. As mentioned before, the mixed RPP differs from the other representations of this problem as it is comprised of both directed and undirected paths. This means that the graph G is now comprised of 3 elements, V the vertex set, E the edge set and A the arc set. In order to solve the MRPP the graph is first transformed to the graph $G_R = (V_R, E_R, A_R \cup A_{RN})$ where A_{RN} is obtained by adding arcs (i, j) for each pair of vertices where the distance is equivalent to the shortest distance between the arcs. Thereafter the arcs (v_i, v_j) in A_{RN} are deleted where the cost function c_{ij} is equivalent or larger than $c_{ik} + c_{kj}$ for some k in V_R are deleted. Finally all parallel arcs are removed from the graph. The first solution method used by the authors is a 4 phase method. Phase one involves computing the shortest spanning tree connecting each vertex in the graph. In phase two the connectivity of the graph is improved by assigning directions to all the undirected edges in the graph. With this new directed graph, $G_D = (V, A_D)$, a minimal cost tour is found in phase 3 and the solution improved in phase 4. This method will give a good solution for relatively small problems. For larger instances a tabu search is used to approximate a solution.

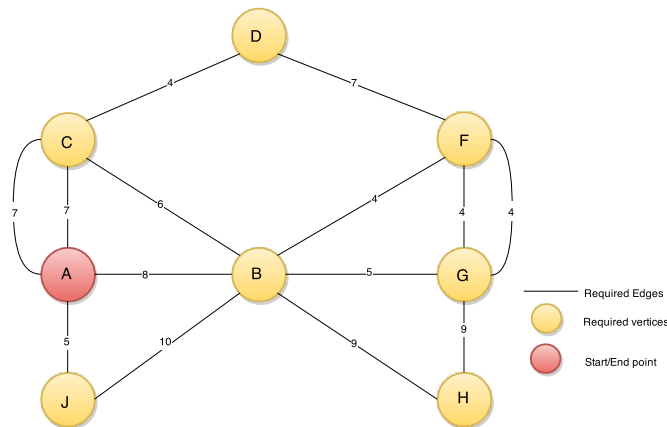


Figure 2.2: Transformed graph with required edges and even degree nodes

2.3.2 Solutions to windy Postman Problems

On the graph $G = (V, E)$ with costs c_{ij} and c_{ji} associated with each edge (v_i, v_j) in E , the windy postman problem can be described as having to find a windy postman tour that transverses the edges $E_R \subset E$. The WRPP is NP-hard therefore while exact solutions do exist for the problem, heuristics need to be used on sufficiently large problems to approximate good solutions. Benavent et al. (2007) introduce 4 heuristics to solve the WRPP. The heuristics are built upon the principles described in Win (1989) and involve transforming the graph $G = (V, E_R)$ into a Eulerian graph and applying Wins algorithm to find the optimal path.

For the WRPPZ, an extension of the WRPP that includes zig-zag services, the arc set in the graph $G = (V, A)$ is subdivided into 4 subsets. These represent non required arcs, arcs requiring single and double service and arcs where zig-zagging is allowed. Irnich (2005) proposed an integer linear programming approach to this class of problem.

A new type of WRPPZ is introduced in Irnich (2008) called EXT-WRPP. It differs in that it takes into account street segment sides, turn restrictions, clusters, public transport and 'knock-off' after last delivery. Street segment sides refer to the fact that a postman will have to travel on either the left or right hand side of the road. If the planner doesn't want the postman crossing roads at major intersection turn restrictions can be introduced into the model. Clustering of arcs help postman effectively sort mail in order of the clusters they will visit along their tour. As described in the introduction, sometimes postmen need to be dropped off at the start point of their tour, this can be modelled as the postmen using public transport (for both their journey to and from their routes.). Finally 'knock-off' is used to model the tour as an open tour. Irnich (2008) showed that this type of problem can be modelled with a graph theoretic approach by transforming it to a asymmetric travelling salesman problem and solving it using heuristics such as nearest neighbour.

For instances where more than one postman services a route (or k-postmen), the problem can be modelled as a Min-Max K-WRPP (MM-KWRPP). Benavent et al. (2010) solve the MM-KWRPP using the 4 heuristics as described in Benavent et al. (2007) and a multi-start solution combined

with an iterative search procedure to find the optimal tour.

2.3.3 Concerns and Conclusion

Solution methods for general RPPs were introduced as were solution methods for WRPPs, which were identified as the most applicable solution method for the Post Offices needs.

However during the course of the research into WRPPs several problems became apparent to the student. Firstly while WRPPs offer a good method to model several constraints that affect the postmen (as discussed in this chapter) identifying the constraints and deciding on the perfect method to weight them and quantify them would be both too time consuming and beyond the skill set of the student.

Secondly while there has been a growing trend of research into arc routing, the models seem to be becoming more complex and specialised. Corberán and Laporte (2015) recognised this in their book titled ‘Arc routing: problems, methods, and applications’ stating that “the state-of-the-art methodologies are becoming increasingly difficult to reproduce and to teach. We may even have reached a point where it may be desirable to devise simpler algorithms, even if this is likely to result in a small loss of accuracy”. This can prove to be problematic as one of the requirements for any solution for the Post Office should be flexibility and ease of teaching. The chosen model should be able to be changed where necessary and with relative ease by people in depots in order to help the Post Office carry out its duties. A solution method therefore had to be found that was both well researched in literature, showed the applicability of arc routing methods to the Post Office and that was easily customisable when needed. The Capacitated arc routing problem was therefore identified as a possible method.

2.4 The Capacitated arc routing problem

The capacitated arc routing problem (CARP) differs from the other solution methods described so far in that it is modelled on a graph $G = (V, E, C, Q)$ where V represents the vertex set, with the depot represented by the vertex V_1 , and E representing the edge set. Each edge in E has an associated demand and cost of service represented by Q and C respectively. There are a set number of vehicles, k , with a positive capacity W . “The objective of the CARP is to find a number of tours such that 1) Each arc with positive demand is serviced by exactly one vehicle, 2) The sum of the demand serviced by those vehicles does not exceed W , and 3) The total cost of the tours is minimized” (Wøhlk, 2008).

There are several variants of the carp, chief of which are:

CARP on Mixed graphs While the classic CARP is modelled on an undirected network, some applications might require the modelling of one way streets for accuracy. The Mixed CARP as studied by Mourão and Amado (2005) could be used to model the routes of motorbikes at the Post Office which must travel on traditional street networks. Belenguer et al. (2006) shows the flexibility of the MCARP, allowing for it to be solved to include prohibited turns, cost limits for trips, windy edges and intermediate facilities.

CARP on Directed graphs Similar to the Directed RPP and Directed CPP the CARP on directed graphs as studied by Maniezzo (2004) is an instance of the CARP where all streets

are modelled as arcs.

CARP with alternate objective functions It might be necessary to minimize factors other than the total distance travelled, as with other arc routing problems. Ulsoy (1985) solved the CARP where there were multiple vehicles with multiple capacities and a cost associated with using each type of vehicle. Here the objective was not to only minimize trip length but also the cost of travel and the cost of vehicles used.

Multi depot CARP The multi Depot carp (as studied by Kansou and Yassine (2010), Liu et al. (2010)) differs from the traditional CARP in that each vehicle is located in one of several depots and can begin and end its journey in any depot. An extension of this is the CARP with Intermediate facilities, where there is one depot but several nodes act as intermediate facilities. Willimse and Joubert (2012) presented a method that used this to model the routing of rubbish removal, where intermediate facilities acted as temporary dump sites. If the intermediate facility is not stationary (e.g a refill car that can come out to you) then the problem is modelled as a CARP with refill points as studied by Amaya et al. (2007).

Periodic CARP Situations, such as weekly newspaper delivery or monthly bill deliveries give rise to easily predictable periodic demand. The planning of delivery routes for such demand is completed by modelling the problem as a Periodic CARP as shown by Lacomme et al. (2005).

2.4.1 Relevance to the Post Office

The Mixed CARP just like the Mixed RPP and Mixed CPP forms the backbone of any solution method that will be looked at due to its relevance to the Post Office. The CARP with alternate objective functions allows for the planning and optimization of postman routes for bicycles, walking postmen and motorbikes. While the Multi Depot CARP, CARP with Intermediate facilities and CARP with refill points allows for the modelling of time bags (the chosen method depending on the depots individual needs). The periodic CARP allows for the planning of mail that has a predictable demand, ensuring that time is not wasted by postmen walking routes that don't have sufficient demand.

With the relevance of the CARP variants established, the solution methods available for the CARP will now be discussed.

2.5 Solution Methods for the CARP

Several heuristics and metaheuristics exist for the CARP and its variants. One of the earliest heuristics introduced was CARPET by Hertz et al. (2000) which approximates a solution by first encoding a list of nodes that represent both serviced and deadheaded edges in a route. The route is then improved using 7 procedures, shorten, drop, add paste, cut, switch and postopt.

Lacomme et al. (2001) presented a solution using a memetic algorithm that modelled giant RPP routes as chromosomes. The solution is then improved and a splitting procedure applied that separates the giant tour into several feasible ones. Three procedures are then applied to the new routes to improve them further, the first involves a relocation of one or 2 consecutive

edges, the second requires two consecutive edges to be exchanged and the third is a two opt move. Lacomme et al. (2004) further extended the MA to include turn penalties, time windows and capacity constraints, maximum trip limits and forbidden turns. The authors found the method to work twice as fast as CARPET is solving benchmark problems. Beullens et al. (2003) modified the memetic algorithm of Lacomme et al. (2001) by allowing the capacity of each vehicle to be exceeded and by introducing a new improvement technique called dir-opt. This method looks at each arc in a route and reverses its direction of traversal in order to see if costs are lowered.

Maniezzo (2004) solve the DCARP by first converting the problem into an asymmetric capacitated vehicle routing problem¹. One of 3 algorithms (a variable neighbourhood search, a genetic algorithm and a multi-start variable neighbourhood search) is then used on the transformed problem in order to approximate a solution.

Amberg et al. (2000) provide a solution for the CARP with multiple depots by converting the problem into a special capacitated minimum spanning tree problem. A solution is then approximated using a constructive heuristic and a metaheuristic that exchanges nodes in a method similar to Lacomme et al. (2001).

Ghiani et al. (2001) solves the CARP with intermediate facilities. The solution is approximated by creating a route made up of several segments, the first from the depot to an intermediate facility, the second linking 2 intermediate facilities if necessary, and the final segment linking an intermediate facility to the depot. Ghiani et al. (2004) then modified the CARP with Intermediate facilities to include maximum trip constraints to the routes generated, allowing for the modelling of shifts and working hours.

Amaya et al. (2007) Solves the CARP with refill points through an integer linear model. The CARP-RP differs from the CARPIF in that the service vehicle/postman would be replenished by a refilling vehicle along their route, with the vehicle returning to the depot afterwards. This method is more flexible than the CARPIF in that refill points are not stationary and can change depending on the inputs to the problem.

Lacomme et al. (2005) solves the Periodic CARP by giving every required edge in the graph a service frequency. A memetic algorithm was proposed that modelled giant RPP tours as chromosomes, and then split according to frequency constraints. These subroutes are then optimised using a daywise procedure.

While metaheuristics such as CARPET and the memetic algorithm provide a good solution, concern has been raised over the time taken to approximate a solution using them. Santos et al. (2009) used several metaheuristics and heuristics to solve benchmark problems and noted that while metaheuristics produced better results, the total time taken to solve the problems increased by factors as large as 101.62. The authors noted that the change in solving time could be noticed from problems as small as 140 nodes and 190 edges. The pilot depot introduced in Chapter 1, alone services close to 800 streets raising concern over the computing power and time needed to use a metaheuristic. The authors also noted that greedy heuristics and more importantly path scanning heuristics are superior to common metaheuristics in that they are simple to encode and

¹The objective of the vehicle routing problem is to find a set of routes for a fleet of vehicles with a known capacity in order to satisfy a customer's needs or a cost function (Laporte et al., 1986). Similar to the CARP, the asymmetric capacitated vehicle routing problem as studied by Vigo (1996) and Laporte et al. (1986) seeks to find routes for a fleet of vehicles with uniform capacity without exceeding the capacity of each vehicle. Vehicle routing problems will not be discussed in this project.

modify allowing planners the chance “to accommodate various real-world specifics of the underlying problem” (Santos et al., 2009).

The path scanning heuristic introduced by Golden et al. (1983) approximates a solution for the CARP by calculating the edge length, edge demand, distance to depot and distance to the next unserved edge. The authors propose using 5 tie break rules to select the next arc to service, the rules are:

1. Maximize the cost from current edge to the depot.
2. Minimize the cost from current edge to the depot.
3. Maximize the ratio of (cost to service)/(demand).
4. Minimize the ratio of (cost to service)/(demand).
5. Use rule 1 if the vehicle is less than half full, else use rule 2.

Tests were run using each selection criteria individually and rule 5 was found to generate the best solution.

Algorithm 1 shows the basic steps of the path scanning heuristic.

Algorithm 1 Path scanning heuristic

Input : Set of required edges and arcs, depot location, capacity of vehicles

Output : Optimal tour

Step 1 Create a set K of the required arcs closest to the unconnected node of the most recent edge serviced.

Step 2 Eliminate all edges from K that if serviced would close the route. This step can be ignored if selecting the edge makes K an empty set.

Step 3 If K has more than one edge and all invalid edges have been removed as in step 2 run a tie break rule to select the edge to service.

Step 4 If K is an empty set, then close the route by selecting the shortest route to the depot.

Several modifications to the path scanning heuristic exist. Pearn (1989) modified the path scanning heuristic by modifying the tie break rules such that they were selected at random. The change allowed for far more solutions to be generated which increased the chances of finding better routes. Belenguer et al. (2006) did not use tie break rules in deciding which edge to service, but rather selected one of the tied edges at random. When the algorithm was used to solve common benchmark tests, the authors found that while the solutions were not optimal as the work of Golden et al. (1983), the differences were negligible. In fact the solutions provided by most modern heuristics and metaheuristics have been shown to be very close to optimum as noted by Corberán and Prins (2010) and could make selecting a specific method an arbitrary step.

2.5.1 Performance of the heuristics

Corberán and Prins (2010) note that most solution values found by heuristics and metaheuristics have in fact proven to be near optimal. The authors go on to state that of the original 3 sets of benchmark tests available to test solutions, *gdb*, *val* and *egl*, all of *gdb*'s instances have been solved optimally, 28 out of 36 of *val*'s instances have been solved while *egl* remains resistant.

This shows the effectiveness of most modern heuristics and metaheuristics at solving routing problems and brought a degree of confidence in selecting the CARP as an applicable solution method to the Post Offices problem.

2.5.2 Conclusion

The CARP and its variants were introduced as well as their solution methods and their potential application to solve the Post Offices problems were also discussed.

Modelling the problem as a CARP was also deemed advantageous in that it could be modified to solve several other problems that the Post Office might have. For example, the Post Office needs to decide on which streets must be serviced by a particular depot before they can begin planning the routes. The Multi Depot CARP can generate solutions that incorporate both activities, allowing not only for time savings in the planning process, but larger more holistic routes on a city/town level.

The scope of the problem and the effectiveness of the solution methods available gave the author confidence to select it as the applicable solution method.

Due to the fact that both heuristics and metaheuristics produce similar solutions to benchmark problems, the final decision on the exact method used to solve the CARP was left up to 1) ease to learn, 2) ease to implement, 3) applicability to the Post Office and 4) quickness of solution. With this in mind and the results from the literature review the path scanning heuristic was chosen as the most applicable solution method for the CARP.

However before proceeding with the solution method, research into off the shelf software alternatives had to be conducted.

2.6 Alternative solutions

It would not be wise to either design or recommend a solution method to the Post Office without looking at alternative software available in the market. Not only does third party software avoid the effort required to ‘reinvent the wheel’ but it is also advantageous in several ways. Firstly implementing the software would be much easier as the developers would have had experience working with similar clients and would know the challenges that would be experienced in an endeavour of this sort. Secondly the support a company offers for troubleshooting problems makes identifying and fixing errors much easier. Finally the company that provides and implements the software/solution would have a measure of liability to ensure its success, as opposed to completing the project independently where all liability lies on the Post Office.

However purchasing software would require significant capital outlay, in order to train staff, get the rights to use it, upgrade systems to accommodate the software and to test the effectiveness of its outputs. Furthermore the software, which may have been built for developed communities might not be suitable for the diverse landscape of a developing country. All of these factors and several more need to be considered before deciding on a method.

Looking at the current market several software alternatives are available, some of which are discussed below:

Route smart technologies RouteSmart Technologies (2015) offers a variety of products to han-

dle the routing needs of most companies. Of these their postal and parcel delivery service would service the Post Office. The technology has been used to optimize the routes of 2 national carriers (the Swiss Post Office and the Isle of man Post Office) and is undergoing small scale implementation by the Unites States Postal service which is experimenting with dynamic routing. The software allows for the generation of multiple routes, for multiple modes of transport specializing in high density areas.

QuintIQ QuintIQ (2015) specialises in resource and task optimization when creating routes. Their postal delivery service is not as advanced as their other offerings and is based mainly on parcel delivery.

Click Software Click Software (2015) offers routing solutions, but like QuintIQ they do not have a dedicated mail delivery solution but rather a parcel delivery solution. What differs them from the other products available is that they allow for the generation of routes focused on alleviating budgetary constraints, which is key to the Post Office.

RouteXL RouteXL (2015) uses a travelling salesman problem to calculate efficient routes. This is combined with live traffic data in order to avoid delays.

Route planner Mapquest Route planner (2015) offers route planning for parcel delivery. The software is tailored for small businesses and was the cheapest out of all the software packages identified in this report.

2.6.1 Conclusion

This report will focus on creating a routing solution from scratch and not on the alternatives listed above, for several reasons. Firstly, given the financial position the Post Office is currently in, there is no appetite by management for any capital intensive projects that don't have guaranteed outcomes. Secondly if third party software is chosen, there still needs to be an in-depth understanding of the way it works in order to both make and informed choice on the software to use and to deal with any problems that might arise during implementation. Therefore by focusing on the theory behind and providing a working example of the concepts of arc routing, this report will give the Post Office the chance to either build their own system or make an informed decision on a stand alone software package.

2.7 Conclusion

Several solution methods for arc routing problems have been described in this chapter. Firstly the traditional CPP was introduced from which the rest of the solution methods in this chapter have evolved. However it is not flexible enough to represent the routing problem described in this report as edges need to be modelled in the graph. While the RPP and its variants are adequate to model this problem, the difficulty to implement the solution makes it both unrealistic for a final year project and as a easy to implement solution for the Post Office. CARPs however do provide relatively simple to implement solutions, that are customisable and that are supported by a plethora of literature. Modelling the solution as a CARP gives the Post Office the chance to build upon whatever solution is proposed, and if seen fit, change the solution to incorporate intermediate facilities, time windows etc. The remainder of this document will focus on the solution design as well as the results of several tests done on the proposed solution method before presenting the final solution.

Chapter 3

Solution Design

3.1 The chosen method

This chapter describes the chosen solution method in detail. Algorithm 2 shows the overall steps to implementing the path scanning heuristic as was explained in the literature review, each step will be discussed in this chapter.

Algorithm 2 Complete path scanning heuristic

Input : Set of required edges and arcs, depot location, capacity of vehicles

Output : Optimal tour

Step 1 : Transform the graph to one comprised entirely of directed arcs.

Step 2 Create a set K of the required arcs closest to the unconnected node of the most recent edge serviced.

Step 3 Eliminate all arcs from K that if serviced would close the route. This step can be ignored if selecting the arc makes K an empty set.

Step 4 If K has more than one arc and all invalid arcs have been removed as in step 2 run a tie break rule to select the arc to service.

Step 5 If K is an empty set, then close the route by selecting the shortest route to the depot.

3.2 Representing the problem on a graph

Like most arc routing problems, the first step in solving the MCARP is transforming the graph $G = (V, E)$ into a simplified version in order to approximate the solution using a heuristic. For this problem the graph was transformed into a fully directed graph $G = (V, A)$ by replacing every edge $(v_i, v_j) \in E$ by two arcs, corresponding to the two possible directions an arc can be traversed. These new arcs $((v_i, v_j), (v_j, v_i))$ are identified by indices's. Each arc has a demand, a service cost (the cost to deliver mail on a street) and a travel cost (the cost to deadhead a street).

As per all other arc routing problems described in this report, the arc set A is split into a subset of required arcs $A_r \in A$ representing a set of required arcs . Arcs that are not required, do not have a service cost or a demand but do have a travel cost in case they are deadheaded. Each problem set will also have one depot, D , form which a set, K , of vehicles with a limited and

equal capacity C will begin their journeys. However as seen in the description of the Post Office, mail delivery is carried out using three forms of transportation, walking, bicycles and motorbikes. Because our pilot depot only has 1 route serviced by walking and 2 by motorbike out of 35 routes, it was decided to focus only on the bicycle routes for simplicity.

Now that the graph on which this problem will be modelled has been described, the method used to calculate the capacity of a postman and the demand on a street will be discussed

3.3 Determining the capacity and demand

Initial assumptions were that demand would be calculated using the number of delivery points on a street, while the capacity of a postman would be limited to the number of letters that can fit into a satchel/bike basket. However after meetings with the Post Office and having visited a pilot depot, this quickly proved to infeasible due to several reasons chief of which was that the postmen do not have a capacity. While there is a limit to the mass of mail a postman can take at any one time, if a days mail exceeds the mass Post Office staff simply pack excess mail in a satchel (called a time bag) and drop it off along the postman's route to be collected during his walk.

It would be pertinent to take a pause here and discuss the time bags. It was considered to model the time bags as intermediate facilities, however the main problem was that the location of and the number of time bags was not set. Both variables depended on the mass and thickness of mail that needed to be delivered on the day (e.g. a lot of magazines that may come on the first of the month would take up more space in the postman's satchel and he would require a time bag for the excess mail. A possible work around could be forcing the use of time bags on the routes in predetermined locations where time bags are usually necessary. However the Post Office does not keep a record of days time bags were used, the location of time bags or the number of time bags used. Therefore it was decided that modelling time bags as intermediate facilities was not viable.

Returning to the problem of determining capacity and demand, an alternative was needed that was both viable and met the Post Offices needs. After further consultation with the Post Office, it was determined that their main concern was ensuring that the postmen got as close to 40 hours of work a week as possible. Using this as a starting point, it was decided to model both capacity and demand using time. Capacity was calculated by splitting the 40 hour work week into 5, 8 hour days. An hour was allowed for the morning meetings the depot has, as well as for time spent preparing the mail and planning the routes. A second hour was allowed for any miscellaneous activities that might occur during the day including bathroom breaks and cleaning up of the work station at the end of the shift. These figures were calculated after time sheets at the pilot depot were analysed. This left each postman a total capacity of six hours or 360 minutes a day to deliver mail.

Next average cycle speeds to both service a street and deadhead a street were calculated (3 km/h and 7 $km/hour$ respectively) by simulating the activities and measuring the total time taken for each. With this and using the length of a street, the total time required to both service and deadhead the street could easily be calculated, this time value will be used to input the service and travel costs. Finally the total time demand on the street was represented as the total time needed to service the street.

3.4 Interpreting the solution

Solutions to the CARP are calculated in the form of a list of k bicycle routes ($R_1 \dots R_k$). Each route will have a list of tasks that represent the service of a street. However because the streets are indexed for the CARP, a list will also be generated that helps decode the given output.

3.5 Extended path scanning

Solutions were generated using the path scanning heuristic. The algorithm works by systematically adding the closest unserved arcs to the end of a trip. This continues until the postman's capacity is exhausted (this would be when the trip would take the postman six hours to complete), the trip is then closed by making the postman visit the depot. This process will continue until every street has been serviced by a postman. The actual algorithm used to route the postmen is based on a waste collection application proposed by Willemse (2010).

Tie break rules are used to decide on the arc to service in the eventuality that there are two or more arcs close to the nearest unserved node. The rules and their respective pseudo code is presented below.

3.5.1 Rules 1 and 2

Rule one selects the arc that will take the furthest trip back to the depot after service, while rule two selects the arc that would create the shortest trip back to the depot after service. The pseudo code presented below is for rule one. Rule two's pseudo code is similar except for the direction of the inequality.

Algorithm 3 Rule 1

Input : nearest-arc-list

Output : arc-to-select

Best-distance = -1

For each arc **in** nearest-arc-list

 distance-to-depot = distance[arc][depot]

if distance-to-depot > best-distance:

 best-distance = distance-to-depot

 arc-to-select = arc

endif

endfor

3.5.2 Rules 3 and 4

Rule three selects the arc that has the highest demand to cost to service ratio, while rule four selects the arc that has the lowest demand to cost ratio. The pseudo code presented below is for rule three. Rule four's pseudo code is similar except for the direction of the inequality.

Algorithm 4 Rule 3

Input : nearest-arc-list**Output** : arc-to-select

best-yield = -1

For each arc **in** nearest-arc-list **if** serviceCost[arc] == 0: arc-yield = 3000 **else** arc-yield= demand[arc]/serviceCost[arc] **if** arc-yield > best-yield:

best-yield = arc-yield

arc-to-select = arc

endif**endfor**

3.5.3 Rule 5

If the postman has already finished half of his deliveries then rule 2 is selected otherwise rule 1 is selected.

Algorithm 5 Rule 5

Input : nearest-arc-list, current-vehicle-load**Output** : arc-to-select**if** (current-vehicle-load \leq capacity/2): **return**(Rule2(nearest-arc-list)) **else: return**(Rule1(nearest-arc-list))**endif**

3.6 Concerns with the tie break rules

As was mentioned earlier in this chapter, demand for an edge was calculated as being equal to the service time of the edge. This affects the tie break rules due to the fact that rules three and four used the demand to service ratio to select an edge. Three workarounds were identified, the first would simply use rule 5 as the tie break rule, while the second option would be to replace service cost with travel cost in the ratios calculated, the final method would calculate demand as the average of the service and travel costs allowing for the tie break rules to be used as intended.

The first workaround is similar to the work done by Golden et al. (1983), who tested each rule of the path scanning heuristic individually and found rule 5 as the most effective.

Corberán and Prins (2010) warn that replacing the service cost of the edge with the travel costs could degrade the overall solution. However it was not stated if this solution would be less efficient than using one tie break rule by itself. Therefore work around 2 and 3 were introduced in order to see the overall effect on the efficiency of the routes. After comparing the results of the tests, a method would be selected.

3.7 Path scanning operators

Path scanning operators were used in conjunction with the tie break rules in order to find the optimal route. These operators tested the arcs/edges and vehicles to ensure that arcs can be added without exceeding vehicle capacity, found the nearest serviceable arcs and updated the solution when suitable arcs were found.

Algorithm 6 Test load

This algorithm checks to see if the total load of the current trip (i.e the time spent by the postman delivering mail thus far) combined with the time demand of the arc will exceed the postman's total capacity.

Input : Arc to be tested

Output : True or False

If tripload + demand(arc) <= capacity

return True

Else

return False

endif

Algorithm 7 Find Nearest Arc

This algorithm looks for the nearest arcs next to the one currently serviced. All identified arcs are appended to the nearest-arcs list with the closest arcs distance stored in the nearest variable.

Input : Previous arc in route

Output : nearest-arcs, nearest

nearest = huge

nearest-arcs = []

For next-arc **in** unassigned-Arcs-list :

 distance-to-arc = distance[previous-arc][next-arc]

if distance-to-arc <= nearest:

 nearest = distance-to-arc

 nearest-arcs = next-arc

else nearest-arcs.append(nextarc)

endif

endfor

Algorithm 8 Check Nearest Arc

This algorithm checks the nearest-arcs list generated in Algorithm 5 and uses Algorithm 6 to test if adding the arc could exceed the postman's capacity. A list of arcs that will not exceed the postmans capacity is then populated.

Input : nearest-arcs
Output : next-arc-fine
 next-arc-fine = []
For arc **in** nearest-arcs
 if test-load(arc)
 next-arc-fine.append(arc)
 endif
endfor

3.8 Building the route

The final stage in the heuristic involved using the path scanning operators and tie break rules to build the route. The algorithms used as well as a brief description of each follows.

Algorithm 9 Add Arc

The Add Arc algorithm searches through a required-unassigned-arc list and for each arc ensures that it isn't housed in a invalid arc list before adding it to the total trip. The arc is then deleted from the unassigned arc list.

Input : arc, nearest, trip
Output : trip
 total-cost += service-cost[arc] + nearest
 trip-cost += service-cost[arc] + nearest
 trip-deadhead += nearest
 trip-load += demand[arc]
 trip-service-costs += service-cost[arc]
 arc1 = required-unassigned-arc-list.index(arc)
if invArcList[arc] :
 arc2 = required-unassigned-arc-list.index(invArcList[arc])
 del required-unassigned-arc-list[max(arc1, arc2)]
 del required-unassigned-arc-list[min(arc1, arc2)]
else :
 del required-unassigned-arc-list[arc1]
endif
 trip.append(arc)

Algorithm 10 Build route

This algorithm uses the add arc procedure from algorithm 9 and the tie break rules to build the route by evaluating each arc in the nearest-arcs list and using the tie break rules if necessary. i.e If the nearest-arcs list has only one arc, the tie break rules aren't necessary, if it has more than one they are and if the list is empty then the tour is complete and the vehicle routed to the depot.

Output : trip

trip-load = 0

trip-cost = 0

trip-dead-head = 0

trip-service-costs = 0

make the first entry in trip = depot

While True previous-arc = trip[-1]

 (nearest-arc, nearest) = find-nearest-arc(previous-arc)

if len(nearest-arcs) == 1

 trip = add-arc(nearest-arcs, nearest, trip)

elif len(nearest-arcs) > 1

 run tie break rules to determine the arc to choose

 trip = add-arc(choosenarc, nearest, trip)

else:

 add depot arc to trip

break

if not required-unassigned-arc-list

 add depot arc to trip

break

endfor

3.8.1 Testing the routes

The next chapter introduces several tests that were done in order to determine the effectiveness of the chosen solution method.

Chapter 4

Testing the chosen method

4.1 Rational for testing

Before presenting the chosen solution method, it was necessary to test the model to ensure its effectiveness. As discussed in the previous chapter there were concerns over the modifications made to the tie break algorithms, furthermore it was decided that the model should be tested against some of the off the shelf software discussed in Chapter 2 in order to see the feasibility of creating an ‘in house’ solution. Finally as will become apparent in this chapter, concerns were also raised over the amount of deadheading in the generated solutions, therefore one final test that replaced all the edges with arcs in the model was performed.

The tests were run on on a 2.7 Ghz iMac with an Intel core i5 processor and 8GB’s of DDR3 memory.

4.2 The test area

The area the pilot depot services was deemed as to large for testing purposes. Therefore a small area within the pilot depots service network was chosen to run all the tests in and is represented by figure 4.1. The total area comprises of 456 edges with a total distance of 90 Km.

4.3 Test 1: Effect of the tie break rules

Due to the concern over the impact changing the tie break rules had (as a result of the demand calculations) on the effectiveness of the output, it was decided to generate routes using the 3 described methods from chapter 3.

To do this 2 walks were selected at random from a list of Post Offices walks for the depot. The walks, which are stored in an excel spreadsheet by the post office (called a PDS), were numbered 1-35 and a rand() function was chosen to pick the two walks.

The two walks that were selected were walks 13 and 3. Figure 4.2 represents the two routes selected to be modelled and the total area included in the model respectively.

The choice of walks also presented an interesting scenario to model. As can be seen from the map in figure 4.1, route 3 is rather short and it could be presumed that this would be for several

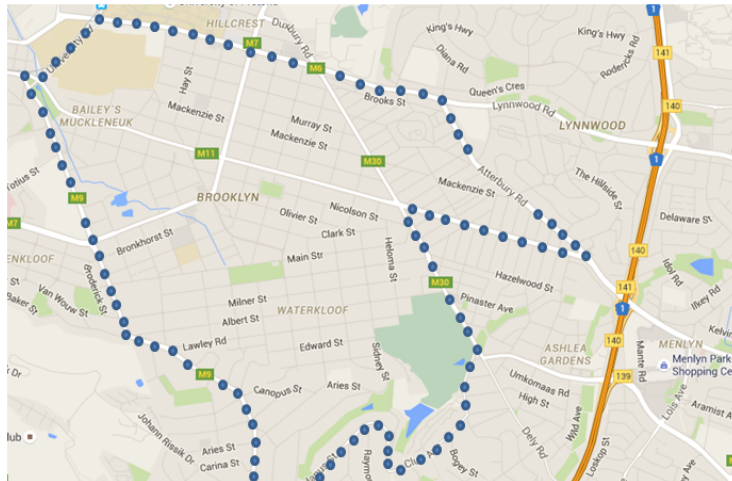


Figure 4.1: Total area that was modelled

reasons. The streets on that route could be serviced rarely or they could form part of the walk of a postman who then visits another street network faraway, making it impractical to model the walks as one walk. Therefore it is clear, even before beginning to the routing activity, that the two routes would be merged. The question however is how would they be merged? Would it be an unrealistic grouping of streets? Would the two would be modelled separately and simply be connected by a single edge? Or would there be one merged and comprehensive route?

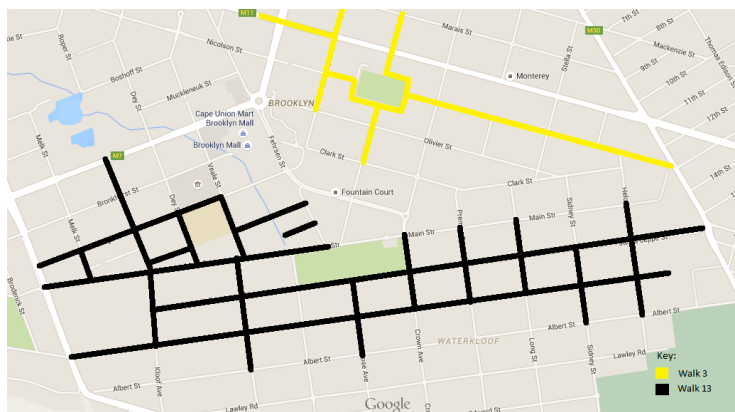


Figure 4.2: Representation of the routes

4.3.1 Scenario 1: Demand as an average

This test calculated the time demand per edge as the average of the total time to service it and to travel along it. It is clearly apparent before even beginning the test that this method is wrong, as this does not show the total time taken to service the street and has the effect of falsely representing the effect of servicing a street on the postman's capacity. However using this method is important in helping analyse the other tie break rules. Since the postman's capacity will not be filled in this scenario and is not used to calculate efficiency or the order of traversal, this scenario helps show

the effect of the proposed changes to the original tie break rules on the order of arcs traversed and by extension the amount of time spent deadheading.

The results of the test are available in figure 4.3. please note that for all images, the Post Office depot is represented by a small house.

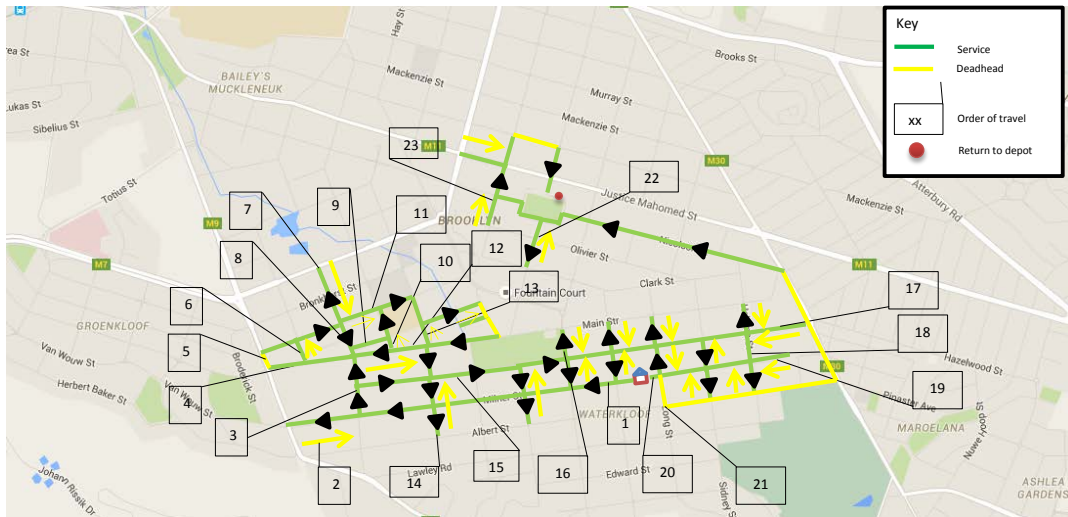


Figure 4.3: The results of scenario 1

4.3.2 Scenario 2 : Demand equal to service time

This scenario calculated the total time demand of an edge as being equal to the total time to service it. Initial assumptions were that the routes would be similar since both routes 3 and 13 would exhaust capacity. However as can be seen from the map there seems to be far more deadheading in the generated route.

Furthermore the order of the roads travelled between scenario 1 and two are also different, analysis of the output showed a 10 percent similarity in the routes, showing a clear indication that replacing service cost by travel cost in the tie break rules had a tangible effect on the routes generated.

4.3.3 Scenario 3: Demand equal to service time using Rule 5 only

Tie break rule 5 was used (and by extension rules 1 and 2) as was shown in Golden et al. (1983). The routes were generated using the new rules and can be seen in Figure 4.5.

4.3.4 Results and considerations

Table 4.1¹ shows the results from the three scenarios of Test 1. As can be seen there was a drop in overall efficiency (calculated as service time over total time) as was expected in literature, due

¹The tables presented in this project have the same format. The total time column first presents the time spent servicing the edges followed by a (s) and then the time spent deadheading followed by a (d). The same applies to the distance travelled column Finally efficiency is calculated (service time)/(Service time + deadhead time)

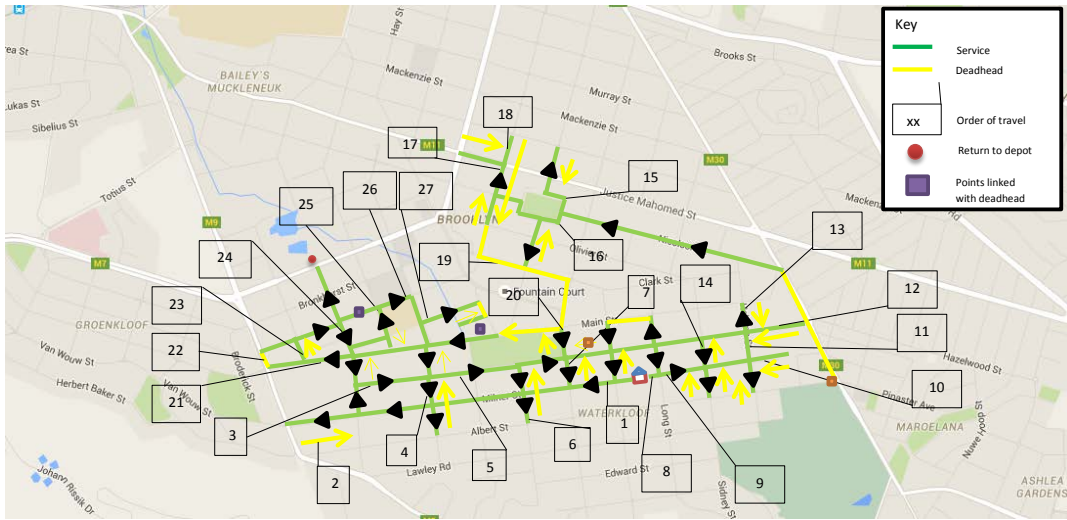


Figure 4.4: The results of scenario 2. Note some deadheading is omitted for graph simplicity. Instead some sections connected by deadheading have been represented by a square on the nodes that are meant to be linked.

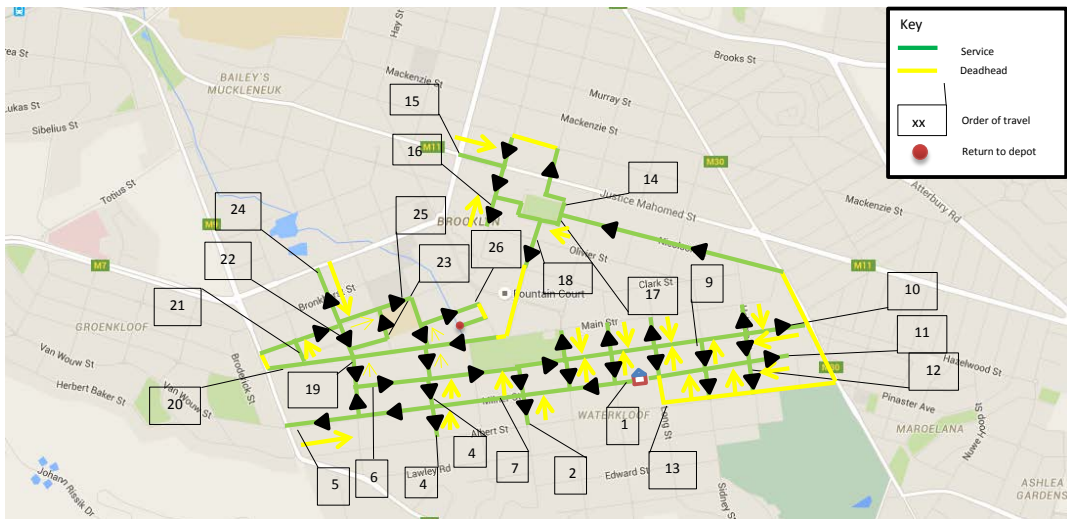


Figure 4.5: The results of scenario 3.

to increased deadheading time. However the overall difference in the total time taken on the walk from scenario 1 and 2 was only 14 minutes (or approximately 2 Km's) and was not large enough to confirm a overall drop in efficiency between the two methods or just a statistical anomaly (the same applies for scenario 1 vs 2 and scenario 2 vs 3). Therefore it was decided to run several more tests, the results of which are available in figure 4.6, in order to see if this trend continued with larger instances of the problem.

Scenario	Total Time (minutes)	Distance Travelled (Km)	Efficiency
Scenario 1	192(s) + 72(d)	9.6(s) + 8(d)	0.72
Scenario 2	192(s) + 86(d)	9.6(s) + 10(d)	0.69
Scenario 3	192(s) + 73(d)	9.6(s) + 8.5(d)	0.72

Table 4.1: Results from the first test

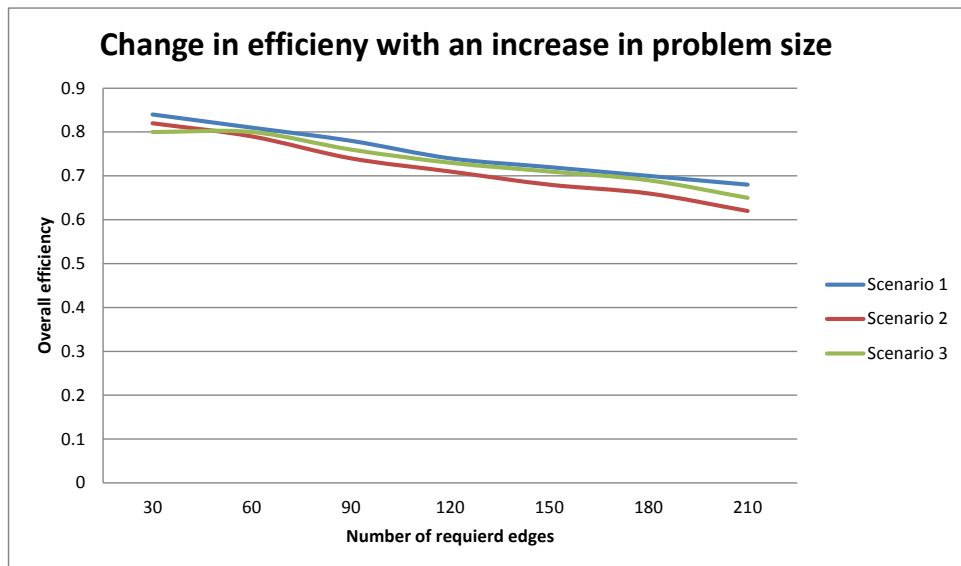


Figure 4.6: Comparison of the overall efficiency of scenarios with an increase in problem size

As can be seen from the figure, the overall trend in the differences between the efficiencies of the three scenarios continues with larger test cases, however it still remains comparatively close in terms of percentages. However with larger test cases a 5% change in overall efficiency can result in huge amounts of time lost deadheading streets.

What is very clear from looking at the routes generated was the amount of deadheading required on already serviced edges. Figure 4.7 shows the effect deadheading has on overall efficiency. Due to the fact that overall efficiency was calculated as (time servicing a street)/(time servicing a street + dead heading time) it becomes apparent that an increase in deadheading decreases overall efficiency and means the Post Office is paying postmen for time spent not servicing a street.

A possible way to avoid this would be to model each edge as 2 required arcs (to model the even

and odd street numbers on each street) even before the graph transformation.

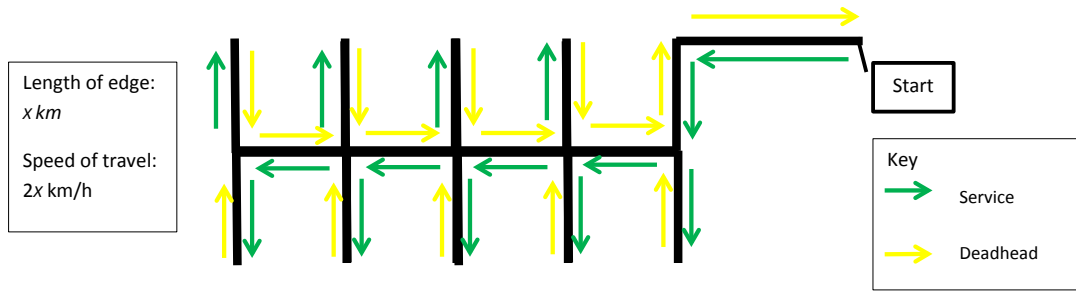


Figure 4.7: Effect of dead heading on a serviced edge. Replacing the edges with arcs would result in the efficiency of the route increasing from 50% to 100%

4.4 Test 2 : Modelling with only arcs

This test sought to see the effect of modelling each edge as 2 arcs on the total amount of deadheading. However attempts to run the test proved unsuccessful due to several reasons chief of which were:

- For large routes modelled solely as arcs, the model would freeze and have to be restarted. This was attempted on 3 PC's and returned the same errors.
- All experiments of smaller models yielded results that were not sufficiently different from modelling the problem with edges.

It was concluded that a more effective test would've been modelling the problem as a Dedicated-CARP and comparing those results with the path-scanning heuristic's results. However due to time constraints this could not be completed and will be left for future work.

A much simpler workaround was to take all deadheaded edges on a road that was serviced and model them as arcs. This was completed on the routes generated in Test 1 and the results are available in Table 4.2.

Scenario	Total Time (minutes)	Distance Travelled (Km)	Efficiency	Change in Efficiency
Scenario 1	192(s) + 8(d)	9.6(s) + 1(d)	0.96	0.24
Scenario 2	192(s) + 9(d)	9.6(s) + 1(d)	0.95	0.26
Scenario 3	192(s) + 10(d)	9.6(s) + 1(d)	0.95	0.23

Table 4.2: Results from the first test with all deadheaded required edges replaced by arcs. This was done by analysing the maps drawn, removing all deadheading on a serviced edge and replacing it with two arcs.

4.4.1 Results and Considerations

Table 4.9 showed the effect of replacing all edges that have a service and a deadhead with an arc. The overall increase in efficiency has shown that all generated routes for this project must be replaced with arcs where dead heading occurs.

However the overall effectiveness of the model has still not been evaluated. It was decided to test the models generated routes against some of the off the shelf software identified in the literature review.

4.5 Test 3: Off the shelf software

This test sought to compare the performance of the model to off the shelf software. This would help the Post Office decide if creating a routing system in house would be as effective as simply buying software.

There were limitations to this experiment however. The high cost of licences to use the software coupled with the limited capacity of free licences meant that a full Post Office walk could not be evaluated. Instead 18 edges were selected at random to represent required edges. Figures 4.5, 4.6 and 4.7 show the routes generated through RouteXL, route planner and the model respectively.

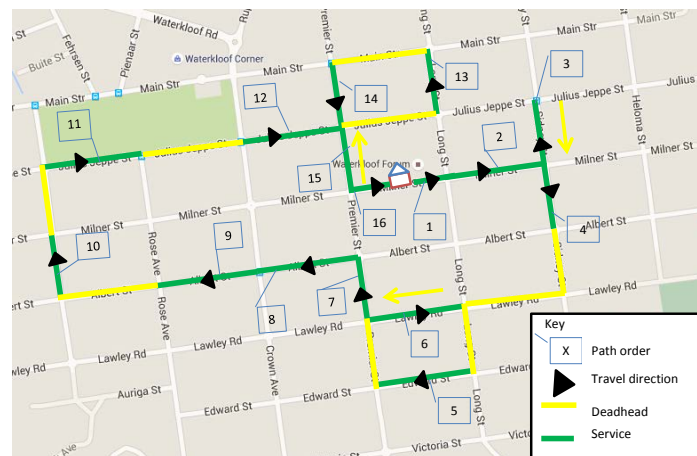


Figure 4.8: RouteXL generated route



Figure 4.9: Route Planner generated route

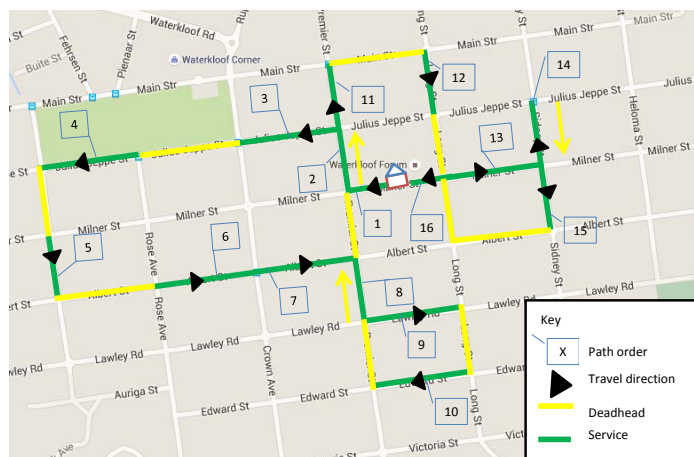


Figure 4.10: Model generated route

4.5.1 Results and considerations

Scenario	Total Time (minutes)	Distance Travelled (Km)	Efficiency
RouteXL	67(s) + 12(d)	9.6(s) + 1.4(d)	0.84
Route Planner	67(s) + 21(d)	9.6(s) + 2.45(d)	0.76
Model	67(s) + 18(d)	3.35(s) + 2.1(d)	0.78

Table 4.3: Results from the third test

As seen in Table 4.4 the route generated by RouteXL was the most efficient while the model and Route Planner produced similar results. Tests done with different required edges produced similar results, as shown in Figure 4.11.

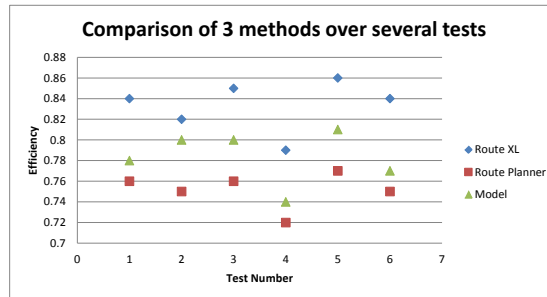


Figure 4.11: Result of several tests

4.6 Conclusion

Small scale tests (limited in edge number did span varying distances) of the model against off the shelf software showed that the model performed to similar standards. The tests however need to be carried out on a larger scale to improve their accuracy. If the Post Office does decide to implement their own dynamic routing system it would be pertinent to perhaps purchase a trail run of the software to perform large scale tests.

4.7 Additional Considerations: Similarity of routes

Before beginning the solution design, it was presumed that a key marker for success in this project would be the similarity of the Post Offices routes with that of the algorithm. However through research and the experiment results it was realised that this might not be as effective a measure as previously thought due to several factors. Firstly the input data did not include the effect of terrain, incline of streets and general fatigue on the postmen. This information is partly the result of numerous calculations and also partly subjective on the part of the postman router. A probable

workaround is modelling the routes using windy edges and attaching a “fatigue factor” to either the calculation of time demand on an edge or on the capacity of the postman (e.g depending on the number of routes done and the “windy-ness” of those routes, the demand is multiplied by a factor of $1.X$, with $X \geq 1$, or the capacity of the postman is multiplied by a factor of $0.y$, with $y \geq 1$). However greater research is needed in order to find the best possible way to calculate these factors and was left for further research and not included in this report.

4.8 Conclusion

This chapter has shown the effectiveness of the overall solution. Through running Test 1 and Test 2 modifications were made the the way the model was run and interpreted to increase overall efficiency. Test 3 showed that the model generated routes similar to software that is available for purchase. However both the off the shelf software and the model failed to take into consideration the ‘windy-ness’ of the routes (as done by the Post Office) which made benchmarking the models routes with the Post Offices routes impossible.

The next chapter shows the routes generated by the model for the depot and shows how the model can be changed by the Post Office staff to represent changes in real world situations.

Chapter 5

Generating final routes

This chapter presents an evaluation of the final routes generated and the results of different stress tests done on the model. Two sets of routes are presented, the first was the routes the model found to be optimum given the inputs and the second was the routes generated after a change in postmen's capacities.

5.1 Final Routes

Using the PDS provided by the Post Office, all the streets in the delivery network were identified. The length of the streets were measured using Google maps, and the demands calculated as in the solution design. Figures 5.1 and 5.2 show a representation of the most efficient and least efficient routes generated by the model.

5.1.1 Analysing the Routes

Table 5.1 shows the total service time, deadheading and utilisation for the routes generated (a detailed breakdown for each route is available in Table 5.2). Looking at the table it becomes

Number of routes	Total Time (minutes)	Distance Travelled (Km)	Efficiency
21	7115(s) + 1606(d)	355(s) + 187(d) (or 7)	0.81

Table 5.1: Breakdown of all routes generated

apparent that instead of the 32 routes that were expected to be generated, only 21 routes were generated. This can be explained by the fact that a) some of the Post Offices walks were small and naturally got merged into larger walks, as was shown in Test 1 of Chapter 4 and 2) due to the fact that factors such as fatigue were not considered for the solution routes would assume constant walking and cycling speed in determining the total time capacity for a postman. However these results created the opportunity to test the simplicity of changing the inputs to the graph in order to ensure all postmen had enough work.

Route Number	Total Time (minutes)	Distance Travelled (Km)	Efficiency
1	360(s) + 82(d)	18(s) + 10(d)	0.81
2	360(s) + 104(d)	18(s) + 12(d)	0.78
3	360(s) + 76(d)	18(s) + 9(d)	0.83
4	360(s) + 78(d)	18(s) + 9(d)	0.82
5	360(s) + 92(d)	18(s) + 11(d)	0.80
6	360(s) + 76(d)	18(s) + 9(d)	0.83
7	360(s) + 75(d)	18(s) + 9(d)	0.83
8	360(s) + 84(d)	18(s) + 10(d)	0.81
9	360(s) + 79(d)	18(s) + 9(d)	0.82
10	360(s) + 92(d)	18(s) + 11(d)	0.8
11	360(s) + 84(d)	18(s) + 10(d)	0.81
12	360(s) + 85(d)	18(s) + 10(d)	0.81
13	360(s) + 76(d)	18(s) + 9(d)	0.83
14	360(s) + 74(d)	18(s) + 9(d)	0.83
15	360(s) + 69(d)	18(s) + 8(d)	0.84
16	360(s) + 72(d)	18(s) + 8(d)	0.83
17	360(s) + 88(d)	18(s) + 10(d)	0.80
18	360(s) + 74(d)	18(s) + 9(d)	0.83
19	360(s) + 75(d)	18(s) + 9(d)	0.83
20	220(s) + 57(d)	11(s) + 7(d)	0.79
21	55(s) + 14(d)	2(s) + 2(d)	0.8

Table 5.2: Full list of routes generated

5.1.2 Effect of a change in capacity

In order to ensure that there were enough routes for the 32 postmen, the capacity of the postmen was altered. Table 5.3 shows the results of the alterations to the capacity and the resulting number of routes.

While the routes generated did allow for the use of the 33 postmen the total time spent servicing the streets decreased by over a third to 220 minutes. Deciding between keeping the current staff complement or striving towards leaner, more efficient depots might prove problematic and the Post Office would need to consider the possible side effects of such a decision when changing their routing method.

Capacity	Number of postmen	Total time (Km)	Distance travelled	Efficiency
360	21	7115(s) + 1606(d)	355(s) + 187(d)	0.81
300	25	7115(s) + 1624(d)	355(s) + 189(d)	0.81
240	31	7115(s) + 1632(d)	355(s) + 190(d)	0.81
220	33	7115(s) + 1632(d)	355(s) + 190(d)	0.81

Table 5.3: Breakdown of change in capacity test

5.1.3 Change in model size affecting time to run

As discussed in Chapter 1, one of the most important aspects of the proposed solution would be ensuring that new routes would be generated in relatively quick times. for example if a postman calls in sick in the morning, the depot should be able to quickly generate new routes to accompany for his absence.

Therefore throughout the entire project, during both the model testing and route generation, the times taken to solve the problems was recorded and presented in Table 5.4.

Figure	Time taken to run model (min)	Time taken to decode model (min)	Time taken to draw maps (min)
Figure 4.3	9	62	23
Figure 4.4	10	64	20
Figure 4.5	10	58	26
Figure 4.10	3	15	8
Figure 5.1	24 ¹	74	38
Figure 5.2	24	64	30

Table 5.4: Time taken to develop routes

While the total time to generate solutions was relatively short, the greatest bottleneck was the decoding of the generated solution from the algorithm. The six maps presented in this report took close to 8 hours to draw and would not be realistic for a Post Office depot that would need to generate upwards of 21 routes at minimum.

The model needs to be integrated with mapping software to simplify the drawing process, however due to time constraints, this was left for future work.

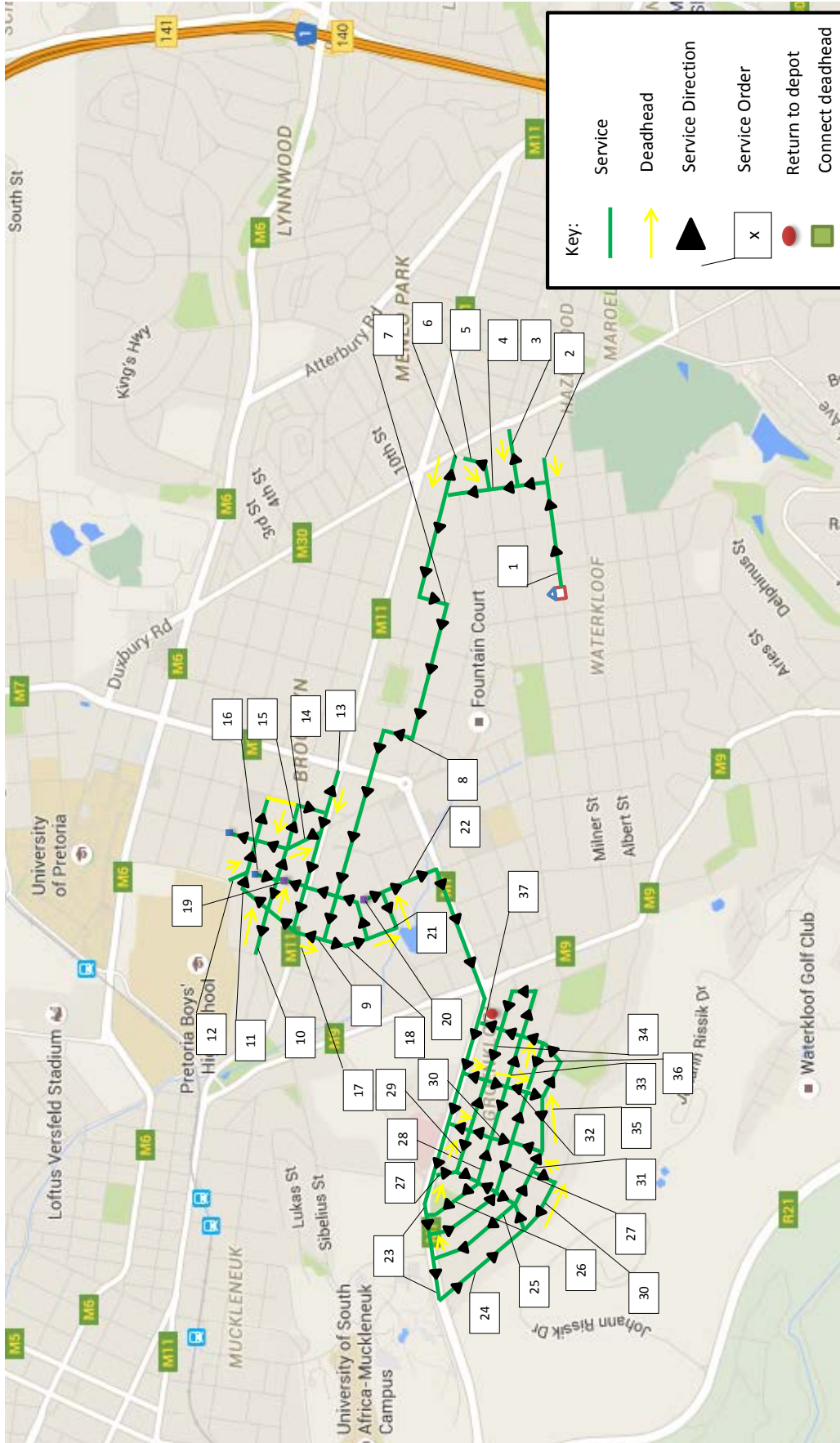


Figure 5.1: Route 20, the least efficient route generated through model. As in Figure 4.4 the connect deadhead blocks are used. The two instances where they are used are in two separate colours for simplicity.

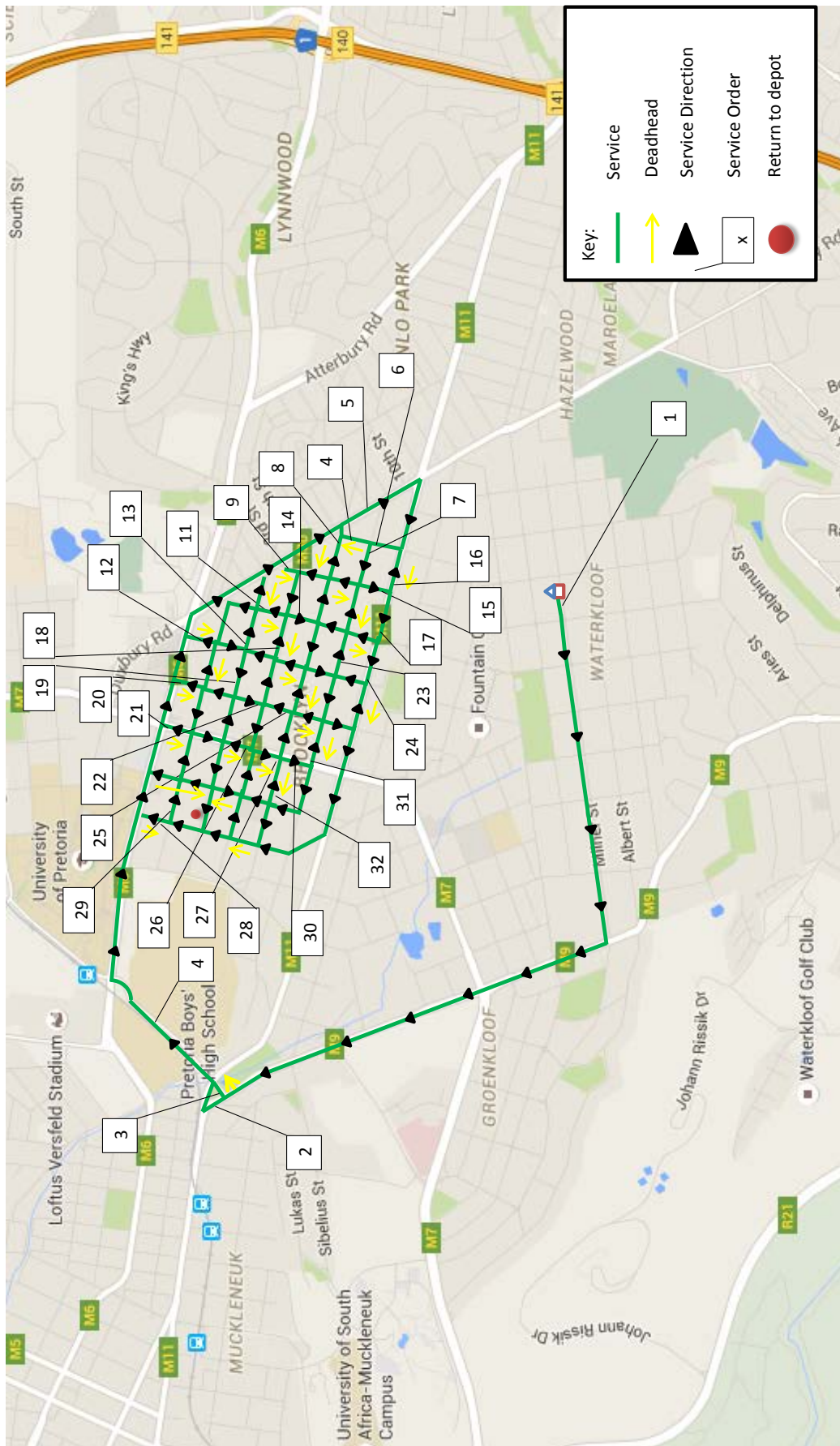


Figure 5.2: Route 15, the most efficient route generated through model

5.2 Conclusion

Routes were generated for the Post Office using the method described in chapter 3. The model did not produce 33 routes for each of the postmen, but rather 21 routes in total. This allowed for the application of a stress test to both ensure show that all postmen can get work and to show the flexibility of the model.

In the opinion of the author, the greatest success was showing that the Post Office can implement dynamic routing solutions at a fairly low cost if the model is created in house.

The greatest surprise was the fact that only 21 routes were generated by the model, which leads to several interesting questions. How does the Post Office balance the need to make the depots more efficient with their workers needs? What is the best possible method to change the work culture of the employees to accept this new method? What is the most effective way to communicate the route to the postmen? i.e Should the depot print out maps whenever they change the routes or should they update the addresses on the shelves shown in Figure 1.3?

This report showed that using a path scanning heuristic to solve the CARP generated routes that were comparable with software that is in use in industry, however greater research needs to be done on the type of CARP necessary to solve the Post Offices problems. The next chapter discusses possible areas of further research to meet all the Post Offices needs and generate a optimal solution.

Chapter 6

Future work

The path scanning heuristic provided good solutions for the postmen routing problem. However the solution cannot be viable if it is not tested against current routes used by the Post Office. Unfortunately, as stated before without taking into consideration windy edges Post Office routes will always appear less than optimal compared to solutions generated with the CARP.

The Min-Max K vehicle windy RPP (MMKRPP) described in the literature review could be a great base to build a final model on, however the relevance of the MMKRPP needs to be compared to the support available in literature as it is still a relatively novel concept that needs further research¹. It is because of this that the CARP will probably remain the most practical method to solve the routing requirements of the Post Office.

6.1 Future work with CARP

Wøhlc (2008) describes the next decade of capacitated arc routing as being focused on two main tasks, 1) improving the size, speed and quality of the results and 2) improving flexibility. The first refers to the work done on both heuristics and metaheuristics that, as new benchmark tests are created and more research is conducted into the field, continue to improve in both the solving time and accuracy of the solutions. Corberán and Prins (2010) note that the growth in arc routing research as of late has given the field its own identity and that this should however help further grow research and interest into the field. However improved solutions need to be balanced with overall flexibility and practicality, as was noted in Chapter 2, Section 2.3.3. Losses in accuracy might be necessary to ensure the model has practical applicability in real world instances.

The two main tasks described by Wøhlc (2008) apply to this project as well. Greater accuracy is needed in the final model to ensure that 1) unnecessary dead heading on edges is eliminated, 2) windy constraints on edges are modelled, 3) multiple vehicle routes can be generated, 4) postmen numbers at depots are optimised and 5) Time bags are modelled . Further research into Multi-vehicle CARPs and CARPs with Replenishment vehicles could result an optimal solution and would form the basis of possible masters level research. Modelling windy edges may prove harder and could complicate the model due to the amount of variables involved. A detailed review of the

¹The two articles that solved the MMKRPP referenced in this report Benavent et al. (2010) and Benavent et al. (2009) are both less than 6 years old, compared to the over 55 year library of research (as claimed by Corberán and Prins (2010)) in arc routing problems in general

Post Offices method for calculating fatigue is needed as is a simpler method of capturing inclines on roads. Once a firm grasp of the method is had, windy factors could be added to the cost to service streets as was discussed in Chapter 4, Section 4.6.1.

However the chosen solution methods must keep in mind that the Post Office also services incredibly rural areas, where the inputs to the model aren't as numerous. The chosen solution method must be flexible enough to work just as well in Pretoria, with motorbikes, foot delivery and bicycles, as in rural KZN with only foot delivery.

Through the students personal experience with this project, a third task is proposed: improving user experience. While this project did succeed in showing that arc routing methods could be implemented to generate Post Office routes it is necessary to improve the way in which the user generates and interprets routes in order to create a viable product for the Post Office. Using online mapping software/GIS software to represent routes and incorporating this with a simple and easy to understand GUI will ensure that the model can be used in real world application.

6.2 Conclusion

This brief chapter represents the possible future work that could be completed to make the CARP solution more feasible. Some of the important decisions that will drive future research have been identified as those that decide on the accuracy of the solution and its flexibility. The solution method presented in future should remain easy to interact with and to modify, to ensure that the Post Office can continually make corrections and improvements if necessary.

Bibliography

- Amaya, A., Langevin, A., and TréPanier, M. (2007). The capacitated arc routing problem with refill points. *Operations Research Letters*, 35(1):45–53.
- Amberg, A., Domschke, W., and Voß, S. (2000). Multiple center capacitated arc routing problems: A tabu search algorithm using capacitated trees. *European Journal of Operational Research*, 124(2):360–376.
- Belenguer, J.-M., Benavent, E., Lacomme, P., and Prins, C. (2006). Lower and upper bounds for the mixed capacitated arc routing problem. *Computers & Operations Research*, 33(12):3363–3383.
- Benavent, E., Carrota, A., Corberán, A., Sanchis, J. M., and Vigo, D. (2007). Lower bounds and heuristics for the windy rural postman problem. *European Journal of Operations Research*, 176(2):855–869.
- Benavent, E., Corberan, A., Plana, I., and Sanchis, J. M. (2009). Min-max k-vehicles windy rural postman problem. *Networks*, 54(4):216–226.
- Benavent, E., Corberán, Á., and Sanchis, J. M. (2010). A metaheuristic for the min?max windy rural postman problem with k vehicles. *Computational Management Science*, 7(3):269–287.
- Beullens, P., Muyldermans, L., Cattrysse, D., and Van Oudheusden, D. (2003). A guided local search heuristic for the capacitated arc routing problem. *European Journal of Operational Research*, 147(3):629–643.
- Click Software (2015). Street level routing. Online: <http://www.clicksoftware.com/optimized-street-level-routing>. Accessed: 29/07/2015.
- Corbera, A., Martm, R., and Romero, A. (2000). Heuristics for the mixed rural postman problem. *Computers & Operations Research*, 27(4):183 – 203.
- Corberán, Á. and Laporte, G. (2015). *Arc routing: problems, methods, and applications*, volume 20. SIAM.
- Corberán, A. and Prins, C. (2010). Recent results on arc routing problems: An annotated bibliography. *Networks*, 56(1):50–69.
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271.

- Edmonds, J. and Johnson, E. L. (1973). Matching, euler tours and the chinese postman problem. *Mathematical Programming*, 88(124):89–124.
- Eiselt, H. A., Gendreau, M., and Laporte, G. (1995a). Arc routing problems, part i: The chinese postman problem. *Operations Research*, 43(2):231–242.
- Eiselt, H. A., Gendreau, M., and Laporte, G. (1995b). Arc routing problems, part ii: The rural postman problem. *Operations Research*, 43(3):399–414.
- Farahani, R. Z. (2012). *Graph Theory for operations reaserch and management: Applications in Industrial Engineering*. IGI Global.
- Frederickson, G. N., Hecht, M. S., and Kim, C. E. (25 October 1976). Approximation algorithms for some routing problems. In *Foundations of Computer Science, 1976., 17th Annual Symposium on*, pages 216–227. IEEE.
- Gass, S. I. and Harris, C. M. (2001). *Encyclopedia of Operations Research and Management Science: Centennial Edition*. Springer Science & Business Media.
- Ghiani, G., Guerriero, F., Laporte, G., and Musmanno, R. (2004). Tabu search heuristics for the arc routing problem with intermediate facilities under capacity and length restrictions. *Journal of Mathematical Modelling and Algorithms*, 3(3):209–223.
- Ghiani, G. and Improta, G. (2000). An algorithm for the hierarchical chinese postman problem. *Operations Research Letters*, 26(1):27–32.
- Ghiani, G., Improta, G., and Laporte, G. (2001). The capacitated arc routing problem with intermediate facilities. *Networks*, 37(3):134–143.
- Golden, B. L., DeArmon, J. S., and Baker, E. K. (1983). Computational experiments with algorithms for a class of routing problems. *Computers & Operations Research*, 10(1):47–59.
- Guerrero, E. (2015). Dynamic routing improves package sorting, delivery. Online: <http://postalemployeenetwork.com/news/2015/04/usps-dynamic-routing-improves-package-sorting-delivery/>. Accessed: 8/11/2015.
- Hertz, A., Laporte, G., and Mittaz, M. (2000). A tabu search heuristic for the capacitated arc routing problem. *Operations research*, 48(1):129–135.
- INFORMS (2015). What is operations research? Online: <https://www.informs.org/About-INFORMS/What-is-Operations-Research>. Accessed: 14-09-2015.
- Irnich, S. (2005). A note on postman problems with zigzag service. *INFOR*, 43(1):33–39.
- Irnich, S. (2008). Discrete optimization: Solution of real world postman problems. *European Journal of Operations Reasearch*, 1(190):52–67.
- Kansou, A. and Yassine, A. (2010). New upper bounds for the multi-depot capacitated arc routing problem. *International Journal of Metaheuristics*, 1(1):81–95.
- Lacomme, P., Prins, C., and Ramdane-Chérif, W. (2001). A genetic algorithm for the capacitated arc routing problem and its extensions. In *Applications of evolutionary computing*. Springer.

Lacomme, P., Prins, C., and Ramdane-Cherif, W. (2004). Competitive memetic algorithms for arc routing problems. *Annals of Operations Research*, 131(1-4):159–185.

Lacomme, P., Prins, C., and Ramdane-Chérif, W. (2005). Evolutionary algorithms for periodic arc routing problems. *European Journal of Operational Research*, 165(2):535–553.

Laporte, G., Mercure, H., and Nobert, Y. (1986). An exact algorithm for the asymmetrical capacitated vehicle routing problem. *Networks*, 16(1):33–46.

Lenstra, J. K. and Kan, A. (1981). Complexity of vehicle routing and scheduling problems. *Networks*, 11(2):221–227.

Liu, R., Jiang, Z., Fung, R. Y., Chen, F., and Liu, X. (2010). Two-phase heuristic algorithms for full truckloads multi-depot capacitated vehicle routing problem in carrier collaboration. *Computers & Operations Research*, 37(5):950–959.

Maniezzo, V. (2004). Algorithms for large directed carp instances: urban solid waste collection operational support. *UBLCS Technical Report Series, Bologna, Italy: University of Bologna*, page 27.

Mapquest Route planner (2015). Route planner. Online: <http://www.mapquest.com/routeplanner/>. Accessed: 13-09-2015.

Minieka, E. (1979). The chinese postman problem for mixed networks. *Management Science*, 25(7):643–6448.

Mourão, M. C. and Amado, L. (2005). Heuristic method for a mixed capacitated arc routing problem: A refuse collection application. *European Journal of Operational Research*, 160(1):139–153.

Pearn, W. L. (1989). Approximate solutions for the capacitated arc routing problem. *Computers & Operations Research*, 16(6):589–600.

QuintIQ (2015). Vehicle routing problem with time windows. Online: <http://www.quintiq.com/optimization/vrptw-world-records.html>. Accessed: 30-07-2015.

RouteSmart Technologies (2015). Routesmart : Postal and parcel delivery. Online: www.routesmart.com/about-routesmart-routesmart-technologies-route-smart/success-ctr-routing-scheduling-software-solutions/success-ctr-postal-parcel-routing-scheduling- Accessed: 30-07-2015.

RouteXL (2015). Routexl. Online: <https://www.routexl.nl/blog/over/?lang=en>. Accessed: 13-09-2015.

Santos, L., Coutinho-Rodrigues, J., and Current, J. R. (2009). An improved heuristic for the capacitated arc routing problem. *Computers & Operations Research*, 36(9):2632–2637.

Thimbleby, H. (2003). The directed chinese postman problem. *Software Practice and Experience*, 33(11):1081–1096.

Tiwari, N. K. and Shandilya, K. (2006). *Operations Research*. PHI Learning.

- Ulsoy, G. (1985). The fleet size and mix problem for capacitated arc routing. *European Journal of operational reaserch*, 22(3):329–327.
- Vigo, D. (1996). A heuristic algorithm for the asymmetric capacitated vehicle routing problem. *European Journal of Operational Research*, 89(1):108–126.
- Wang, H.-F. and Wen, Y.-P. (2002). Time-constrained chinese postman problems. *Computers & Mathematics with applications*, 44(3):375–387.
- Waxer, C. (2014). Digital sos: How technology can save the usps. Online: <http://www.computerworld.com/article/2855146/digital-sos-how-technology-can-save-the-usps.html>. Accessed: 8/11/2015.
- Willemse, E. (2010). *Waste collection routing optimisation*. PhD thesis, The university of Pretoria.
- Willimse, E. and Joubert, J. (2012). Constructive heuristics for the residential waste collection problem. In *ORSSA Proceedings*, pages 19–28.
- Win, Z. (1989). On the windy postman problem on eulerian graphs. *Mathematical Pr*, 44(1-3):97–112.
- Wøhlk, S. (2008). A decade of capacitated arc routing. In *The vehicle routing problem: latest advances and new challenges*, pages 29–48. Springer.