Mail Delivery Problem

Route Optimization with Capacity Constraints

Research Paper
Master Business Analytics

31 Oktober 2013

Jozefien Karskens
Supervisor: Dr. R. Bekker

vrije Universiteit amsterdam
Mail Delivery Problem

Route Optimization with Capacity Constraints

Research Paper Business Analytics
Jozefien Karskens
Studentnr: 2505805
Supervisor: Dr. R. Bekker

31 Oktober 2013
Preface

This Research Paper is a part of the program for my Master Business Analytics at the VU University Amsterdam. The aim of this course is to write a research paper about a subject that is related to business analytics.

This paper is about route optimization for postmen. That is an application in the field of graph theory. Theoretical models and methods for graph optimization will be studied to tackle the practical problem of the postman route optimization.

I would like to specially thank Rene Bekker for supervising and helping me with this research paper.
Summary

The subject of this paper is route optimization for mail delivery. The aim is to determine the optimal route. The route is optimal when it is the shortest possible route but still all the conditions are fulfilled. But the mail delivery situation is constrained by the capacity of the postman. He is not able to carry all the mail at once. That means he has to walk several circuits among his bike to reload mail. But which circuits should he walk to keep the total route length as short as possible?

The mail delivery company Sandd is the inspiration for this paper. They want to have more knowledge about the actual route length of postmen. Besides the mail delivery problem there are several related problems, for example; road gritting, road sweeping, garbage collection and meter reading.

Graphs are in general good representations for route optimization models. In the graph representation are the nodes the intersections and the edges the street sites, or the nodes represent the mailboxes and the edges the distances between the mailboxes. The aim is to find the minimal length tour in the graph taken the route conditions into account. These conditions can roughly be divided into two types the Arc Routing Problems (ARP) and the Node Routing Problems (NRP). In the ARP are the conditions on the edges and in the NRP are the conditions on the nodes, an example of an NRP is the traveling salesman problem.

For the mail delivery problem is the ARP with capacity constraint the best model. This is the so called Capacity Arc Routing Problem (CARP). For the ARP it is easy to obtain an optimal solution by the use of an integer linear programming model. Adding the capacity constraints to the model makes the integer linear programming model very complex because of the exponential growth of conditions. Obtaining the optimal solution for the CARP is NP-hard.

The general conclusion of this research is that it is very complex to obtain the minimal total route length for the mail delivery problem with a limited carrying capacity.

Because it is very complex to derive the optimal route length no implementation have been done. For further research is it suggested to implement some instances of mail delivery problems to see the actual influence of the capacity constraints on the total route length.
# Table of contents

Preface .................................................................................................................. 5
Summary .................................................................................................................. 7
Table of contents ................................................................................................... 9

1. Introduction ......................................................................................................... 10
   1.1. The mail delivery problem .......................................................................... 10
   1.2. Related problems ........................................................................................ 12
       1.2.1. Road gritting ....................................................................................... 12
       1.2.2. Road sweeping .................................................................................... 13
       1.2.3. Garbage Collection ............................................................................. 13
       1.2.4. Meter Reading ..................................................................................... 14
       1.2.5. Paper outline ....................................................................................... 14

2. General problem models .................................................................................. 15
   2.1. Introduction .................................................................................................. 15
   2.2. Arc Routing Problem .................................................................................. 15
       2.2.1. Euler tour ........................................................................................... 15
       2.2.2. Chinese Postman Problem .................................................................. 17
       2.2.3. Rural Postman Problem ........................................................................ 18
       2.2.4. Capacitated Arc Routing Problem ....................................................... 19
       2.2.5. Capacitated Chinese Postman Problem ............................................... 20
   2.3. Node Routing Problems ............................................................................. 20
       2.3.1. Hamilton circuit ................................................................................... 20
       2.3.2. Traveling Salesman Problem ............................................................... 21
       2.3.3. Vehicle Route Problem ......................................................................... 22
       2.3.4. Capacitated Vehicle Routing Problem ................................................ 22
   2.4. Models for mail delivery problem .............................................................. 22

3. Algorithms for mail delivery problem .............................................................. 24
   3.1. Integer linear program for CPP ................................................................. 24
   3.2. Integer linear program for CARP .............................................................. 25
   3.3. Other methods for the CARP .................................................................... 28
       3.3.1. Feasible solutions ................................................................................ 28
       3.3.2. Lower bounds ....................................................................................... 29
       3.3.3. CPP and CARP compared ................................................................. 30

4. Conclusions and further work .................................................................... 31
   4.1. Conclusions ................................................................................................. 31
       ....................................................................................................................... 32
   4.2. Further work ............................................................................................... 32

5. Bibliography .................................................................................................... 33
1. Introduction

This paper is about route optimization for mail delivery. The aim is to determine the minimal total length a postman has to walk in order to deliver all the mail. But the optimization problem is constrained by the capacity of the postman. He is not able to carry all the mail at once, thus he has to reload mail at a reload point. That is possible when his route consists of several circuits along the reload point. But which circuits should he walk to keep the total route length as short as possible?

Route optimization is an often occurring problem. Especially for companies that serve or deliver products at different locations it is a daily challenge to minimize the total traveling distance. In the mail delivery industry optimization plays an important role. This paper is inspired by the question of the postal delivery company Sandd (see Section 1.1). The postal delivery company wants more knowledge about the actual total route length of each postman in his neighbourhood. The aim of this paper is to show the complexity of the route optimization problem with capacity constraints.

In the remaining of the chapter, first, the situation of the mail deliver problem is explained. Second, a selection of similar problems is described to give insight in comparable problems.

1.1. The mail delivery problem

The postal delivery company Sandd is the inspiration for this paper. Sandd is one of the two postal delivery companies in the Netherlands. They only deliver mail from companies, but the mail is delivered at private households and at companies. The postal delivery company deliver twice a week (Tuesday and Friday) around 7 million mail items per day with more than 14,000 postmen [24].

At the postal delivery company the question raises if the planned route lengths for the postmen meets reality. The better the postal delivery company can predict the actual length of the routes the better they can determine the delivery costs. With 14,000 postmen a small difference per neighbourhood can result in a big nationwide difference in total delivery costs. Another feature is that the rewards per neighbourhood can be determined much more precisely if the route length is accurate. The accurate rewards are useful to justify the rewards per neighbourhood to all employees.

The mail delivery process is as follows [23]: the postal delivery company delivers the mail to the postman. The postman puts the mail in the right order according to the route he plans to walk through the neighbourhood. The mail is transported by bike in panniers to the neighbourhood, where the bike is placed on a strategic place. Because the postman cannot hold all the mail at once he has to return several times to the bike. This means that the postman starts at the bike and will pass by the bike several times to refill his delivery bag.

In Figure 1.1 an example of a neighbourhood of the postal delivery company is displayed [23]. The red lines represent the street sides that need to be traversed and the green dots represent all the mail boxes. Note that the demand differs per mailbox. The postman has to traverse every mail box at least once to deliver the mail. It this example it is assumed that traversing each street side once ensures that every mailbox will be served. But because the mail is too much to carry at once the postman has to walk several cycles.
The bike with the mail will be placed at one of the intersections and from that point circuits will be walked through the neighbourhood. Note that it is possible that the postman move his bike to during the delivery. But in this example and in the rest of the paper it is assumed that the bike will not be moved during the delivery. This assumption is made for two reasons. The first reason is that the model becomes much more complex when it is possible to replace the bike. The second reason is that moving the bike takes time and effort of the postman. It is difficult to determine if the time and effort is in advantage of the decrease of the total route length.

In Figure 1.2 an example of a route with 3 cycles is displayed. It is an example of three possible circuits that form a full route; it is not assured that this is the optimal solution. In the route some streets are traversed more than once. The dotted lines indicate that the street side is only traversed and not served in that cycle, which is called deadheading. This example is an illustration of the main problem in the mail delivery problem. The aim of the mail delivery problem is to obtain these circuits with the goal to minimize the length of deadheading and taken into account that the demand of a circuit is not bigger that what a postman can carry.

From a mathematical point of view the main question is: what is the minimum total length of the route for a postman in his neighbourhood, taken into account that every circuit has a constraint capacity and that all circuits start and end at the same point? This is a capacity constraint route optimization problem.
The most common way for route optimization problems is to transform the problem into a graph. In Figure 1.2 is assumed that all the mailboxes on one street side are served in once. With that assumption (what is explained in Section 2.4) the graph representation is as follows. In the graph $G$ the edges $E$ are representing the roads and the nodes $V$ are representing the intersections. The aim is to find the minimal length tour in the graph, taken the route conditions into account.

1.2. Related problems

The mail delivery problem is not the only situation where capacity constrained route optimization problems appear. There are much more practical situations where capacity constraints restrict the route optimization problem. This section describes a selection of often occurring practical problems that are similar to the mail delivery problem. With the background of similar problems it is possible to better understand the context of the route optimization problem.

1.2.1. Road gritting

The road gritting situation, described by Eglese [1], appears in the winter when the roads are frozen and it is necessary to grit the roads with salt to defrost them. The road gritting problem is similar to the mail delivery problem in the sense that a route needs to be determined that serve every road with taking the salt capacity of the gritting vehicles into account. When the vehicle is empty it is not possible to grit the road, the vehicle can just traverse the streets. The aim, as in the mail delivery problem, is to obtain a route with circuits along the depot that minimize the deadheading of the vehicles. By passing the depot the vehicle can reload salt (equal to the postman bicycle location to refill the mail bag). In the gritting problem it is possible that there are several vehicles while in the mail delivery problem there is just one postman. But that does not influence the goal of the problem: minimize the total deadheading in the route. The difference only appears in the implementation, the postman walks all circuits itself while in the road gritting problem the circuits are divided among the different gritting vehicles.

When there are more depots, the situation is slightly different from the mail delivery problem, even when the postman can move his bike. In the mail delivery problem, the route exists of circuits; only the starting point of the circuits can differ. Between the circuits the postman can replace his bike to get a new starting point, and the circuits will than start from that new place. Note that all tours are circuits. In the case of more depots the vehicles can also reload salt at another depot during the day; the vehicles just have to return to the starting point at the end of the day. The aim is still to minimize the total deadheading of the vehicles. But the options for possible routes are extended with paths from one depot to another. It makes solving the problem even more complex (see Chapters 2 and 3 for the complexity of the mail delivery problem. In the papers of Eglese [1] and Li & Eglese [2] heuristics are described to obtain feasible solutions for this multi-depot problem because obtaining the optimal solution is too complex.
1.2.2. Road sweeping

Road sweeping is another daily occurring activity that deals with route optimization. Cleaning vehicles have to sweep the streets to collect the rubbish. All streets should be cleaned once in a while; this means that not every street is swept every day. Also this road sweeping, as described by Eglese & Murdock [3], can be modelled as the mail delivery problem with limited capacity, with the aim to minimize deadheading. The capacity limitation is the amount of rubbish that fits in the vehicle (same as the mail carrying capacity of the postman). But in the road sweeping problem there is another capacity constraint; the maximum hours a road sweeper can work on a day. It means that the solution does not only contain the circuits but also the day that a certain circuit is done. The mail delivery problem gets a solution for one day while the road sweeping problem gets a solution for a time period of more days (weeks, months or years depending on the model).

Another difference with the mail delivery problem is that the locations to dump the rubbish are not at the starting point. The vehicles have to visit the dump locations during the day and at the end of the day such that an empty vehicle returns at the depot. Equal to the road gritting problem that is described in Section 1.2.1 this problem is too complex to solve to optimality. But an approach that can be used is to obtain a feasible solution as described in the paper [3].

1.2.3. Garbage Collection

In 1970 Stricker [15] was the first person that studied the garbage collection situation. Households place their garbage bags along the roads and the garbage collector has to collect all the garbage. The study of Stricker was focused on minimizing the total route length needed to collect all the garbage. The goal of minimizing the total length is the same as the goal of the mail delivery problem, i.e. minimize the total deadheading length. The garbage collection problem is one of the most common practical situations of the route optimization problems with capacity constraints.

The aim is to collect all the garbage in a defined area within a working day. During the day the garbage truck has to dump the garbage, because the size of the truck is limited. The problem is similar to the road sweeping problem in the sense that demand is collecting instead of delivered. Besides emptying the vehicle cannot be done at the starting point but is carried out at other locations. The difference with the road sweeping is that all the streets in one area need to be visited at the same day.

The garbage collection has also some similarities with the road gritting model; in both situations there is more than one vehicle involved. But as already described in Section 1.2.1, having multiple vehicles does not influence the aim of route optimization problem. The aim of the problem is still to minimize the total length of deadheading streets.
1.2.4. Meter Reading

A less obvious representation of a route optimization problem is that of the meter reading problem. Gas companies have to visit every household to read the meter for billing purposes. When the meter readers have efficient routes along the households the total time to read all the meters will be reduced. The problem is briefly described in [15]. The aim is to obtain circuits that minimize the total length such that every household is visited. The routes have to be circuits because all the meter readers has to start at the company and will end there as well to give the outcomes of the meter readings to the company. The aim of this model is the same as the aim of the mail delivery problem; minimize the deadheading. The only difference is that the capacity is restricted by the working hours of the reader and not by physical capacity.

1.2.5. Paper outline

The rest of the paper is outlined as follows. In the second chapter the different theoretical models for route optimization are given. The aim of this chapter is to give more background about the theoretical side of the mail delivery problem. In the third chapter the complexity of the mail delivery problem is introduced by the use on integer linear programming. In chapter three are also some feasible methods explained. In the last chapter the results of the research are presented and some further recommendations are given.
2. General problem models

The capacity constrained mail delivery problem is described in Chapter 1 of this research paper. The descriptions given in that chapter give an idea of the wider range of optimal route planning problems. In this chapter we will look at the more mathematical side of the route optimization problems. In Section 2.1 an introduction is given. In Section 2.2 some models of the so called arc routing problems are described. In Section 2.3 the node routing problem is explained. In Section 2.4 the models are interpreted for the mail delivery problem.

2.1. Introduction

Graphs are in general good representations for route optimization models. The aim is to find the minimal length tour in the graph with taken the route conditions into account. These conditions can roughly be divided into two types of conditions. In the first type of problems all edges $E$, or a subset of $E$, need to be traversed during the tour. This group of problems are called Arc Routing Problems (ARP). In the second type of problems all nodes $V$, or a subset of $V$, need to be visited during the tour, the so called Node Routing Problems (NRP). Within these two different types of problems there are several variants on the conditions and restrictions possible. A selection of the variants is described in Section 2.3 and 2.4.

At the end of Section 1.1 is explained that the mail delivery problem can be represented as a graph. In this representation the edges are the street sides and the intersections are represented by the nodes. In this representation the conditions are on the edges, all edges need to be traversed.

Note that in the rest of this chapter it is assumed that the graphs are undirected unless indicated otherwise. That is because a postman is not bothered by the traffic direction in a street, he can traverse a street in both directions.

2.2. Arc Routing Problem

The Arc Routing Problem (ARP) is the general definition for all route optimization problems that are conditioned by a set of edges in graph $G(V, E)$ that need to be visited during the tour. There are several different variants of this class of problems. In this section some basic and mail delivery problems related variants will be described.

2.2.1. Euler tour

Leonhard Euler was the first person that described the ARP [8] in August 1735. This problem is known as the Königsberg bridge problem. In 1735 at Königsberg there were 7 bridges over a branched river, see Figure 2.1. Euler was wondering if it was possible to walk over the 7 bridges of Königsberg exactly once and end at the starting point. In the case of Königsberg it is not possible; an 8th bridge is needed [9]. The more general question is: is it possible to find in a graph a route that visits all edges exactly once? If the route starts at the same point as the route ends than it is called an Euler circuit, if it ends at another point it is called an Euler path [9].
Euler discovered that it is possible to make a circuit or path if the graph fits some conditions.

A connected graph has an Euler circuit if and only if every vertex has an even degree.

To obtain a circuit the route has to go out of a vertex as many times as it goes into the vertex. This means that every vertex needs an even degree to be a so called Euler graph [10]. A more formal proof can be found in [13]. To determine that a graph has an Euler path the condition is slightly different from the condition for an Euler circuit.

It is possible to find an Euler path if and only if 0 or 2 vertices have an odd degree [10, 13].

If there are zero odd degree vertices than the graph fits the conditions above and so there is an Euler circuit. The number of odd degree vertices is always even in a connected graph, its proof can be found in [10, page 180], so it is not possible to have just one odd degree vertex. Consider an Eulerian graph where all the vertices have an even degree, then there is an Euler circuit in the graph. When omitting one edge there is not a circuit anymore but there still is a path with the points of the omitted edge as starting and end points of the path. These points have an odd degree now, as it started with an even degree in the Euler graph [13]. When more edges are removed to obtain more odd degree nodes the route is not a path anymore, so it is not possible to have more than two odd degree nodes. For a more formal proof see [13].

![Figure 2.1: Köningsberg bridge problem [17]](image)

With the properties described above it is always possible to turn a graph into an Euler graph. By adding edges between odd degree nodes the initial graph becomes an Euler graph. The opportunity to transform graphs into an Euler graph is an important feature that is also used in other models of ARP.
The two properties above only involve total undirected graphs. But these properties also hold for mixed or directed graphs.

It is possible that a graph has an Euler circuit or path if all or a subset of the vertices is directed.

The proof therefor is based on the same idea; the route has to go out of a vertex as many times as it goes into the vertex. Thus it is needed that the number of incoming arcs is equal to the number of outgoing arcs and the undirected edges should be of an even number in every vertex. This makes it possible to go in a vertex as many times as going out of it. For the Euler path there are two extra conditions on the two odd degree vertices. The first condition is that the difference in the degree of incoming and outgoing arcs should not be more than one. The second condition is that one vertex should have one degree higher for the outgoing arcs (start point) and one vertex should have one degree higher for the incoming arcs. This follows from the properties above because the ideas are the same for the mixed situation.

The two described requirements above just identify that it is possible to determine an Euler circuit or path in a graph. These requirements do not specify how an actual route can be determined from an Euler graph. To obtain an actual route the FLEURY algorithm \[9\], that is described many years later than the discovery of the Euler conditions, can be used.

2.2.2. Chinese Postman Problem

The Chinese Postman Problem (CPP) was first suggested by the Chinese mathematician Kwan in 1962 [12]. It is the most fundamental ARP on an undirected graph. The aim is to find the shortest total traveling route length whereby every edge is visited at least one time. The difference with the question of Euler is that Euler was wondering if there exists a circuit while Kwan was wondering what the shortest route is. In the CPP it is possible that some of the edges will be traversed more than once to fulfil the condition that every edge is traversed at least one time.

When the graph representation of the CPP is an Euler graph, then the Euler circuit is the shortest possible route with a length equal to the total length of all edges. It is the optimal shortest route because the route cannot be shorter than the total length of all edges because that is the condition of the CPP. When the CPP graph is not an Euler graph some edges should be traversed more than once to fulfil the condition that every edge is traversed at least one time. By extending the graph with some extra edges the CPP graph can always be transformed into an Euler graph. It is described in Section 2.2.1 why this is possible. The total length of all the edges in this expanded Euler graph is the length of the route.

The aim is to obtain the set of multiple used edges with the shortest total length that extends the initial graph into an Euler graph. Determining this subset of edges can be done by solving a so called perfect matching problem, as described in the article of Derig [12, page 5]. The perfect matching method is based on determining a subset of edges such that every node is connected by exactly one edge of the subset. In this CPP the nodes that need to be connected in the subset are all the nodes
with an odd degree in the initial graph. Extending the initial graph with multiplications of the edges of the subset creates an Euler graph. It is also possible to represent this CPP as an integer linear programming problem; this is described in Chapter 3.

The CPP is basically described on undirected graphs, but this CPP is also investigated for directed and mixed graphs. The complexity for a directed graph is the same as for the undirected variant. Only in the mixed variant it becomes much more complex to solve to optimality.

If a graph is totally directed or totally undirected it is possible to determine the optimal minimal route length in polynomial time.

Finding the optimal solution is basically determining an Euler circuit and that can be done in polynomial time, see [18]. For a mixed graph it becomes NP-hard to obtain an optimal shortest route solution. To obtain an Euler graph the first step is to obtain the incoming and outgoing degree of every node. Because in a mixed graph a part of the nodes has no direction it cannot be determined if the incoming and outgoing degree of a node is equal. Deciding the direction of an undirected node depends on the degrees of all nodes, directed edges and the directions of the other undirected edges. This is not easy to determine. This problem is NP-Hard, for more detail see [18].

2.2.3. Rural Postman Problem

The Rural Postman Problem (RPP) is a variant of the CPP, described for the first time by Orloff in 1974 [8]. The only difference with the CCP is that in the RPP just a subset $R$ of the edges $E$ needs to be visited. All edges in $E$ can be used to determine a route, but just the edges $R, R \subseteq E$, are mandatory to be in the route [15]. This means in the case of the mail delivery problem that streets without mailboxes can be traversed but it is not necessary to traverse them because there are no mailboxes that need to be served. Typically in rural areas are many streets without houses; this is where the name Rural Postman Problem refers to.

The RPP is for the directed and undirected graph NP-hard to solve.

It is pointed out in [8] that a RPP is NP-hard to solve. In the paper [15] is described that it is possible to model the RPP as an integer linear programming model to solve the problem to optimality. Because the constraints grow exponential when the size of the graph increases, is it NP-hard to obtain the optimal solution. But to be able to solve the RPP to feasible solutions many heuristics have been obtained [15]. Most of the heuristics are based on a method that extends the graph with extra edges between the nodes with an odd degree of edges out of subset $R$; this can be done with a
perfect matching problem (described in Section 2.2.2). It is difficult to ensure that this obtained extended graph is the optimal solution.

Only when \( R = E \) it is possible to find an optimal solution in polynomial time, because the problem is then equal to the regular CPP, described in section 2.2.2 [15].

When the RPP is on a mixed graph the problem is called the stacker crane problem, which is also NP-hard [15]. Stacker Crane Problem is the situation whereby some of the edges are directed and some are not. It is not possible to drive over a one-way road in opposite directions, while in a village street that is possible. Because this is not part of the scope of this research more information can be found in [15].

### 2.2.4. Capacitated Arc Routing Problem

The Capacitated Arc Routing Problem (CARP) is the ARP whereby the lengths of the circuits are constrained by the capacity of the traveller. That means that several circuits need to be made not to exceed the capacity of the traveller. This problem is described for the first time by Golden and Wong in 1981 [4].

The CARP problem is related to the RPP as in both problems a tour needs to be determined through a subset \( R \) of the set edges \( E \) and when \( R = E \) the CARP is related to the CPP and is than called the Capacitated Chinese Postman Problem (CCPP, see Section 2.2.5). The only difference is that in the CARP model the capacity of the traveller is an extra constraint on the route. The CARP and RPP (or CPP) are equal if the capacity of the traveller is bigger than or equal to the total demand of all edges in \( R \) [15]. In that case there is no capacity constraint because the capacity of the traveller is bigger than or equal to the total demand in the graph. Note that traversing an edge without serving influences the length of the route, but does not have impact on the capacity of the traveller.

Every edge \( e_{ij} \), where \( i \in V \) and \( j \in V \) represent the nodes where the edge is connected with, has two values. The first value is \( d_{ij} \), the demand on that edge and the second value is \( c_{ij} \), the cost to travel (mostly length of the edge) over the edge. The aim is to obtain the shortest route possible with taken into account that all the vertices with a positive demand (\( d_{ij} > 0 \), that is subset \( R \)) are visited at least once. Shortest route means the route with minimal total length, in other words the sum of \( c_{ij} \) of all the traversed and served edges.

---

The CARP problem is NP-Hard.

That CARP is NP-Hard is shown in the paper of Wong and Golden [4]. They also found that even finding a solution that is less than 1.5 times the optimal solution is NP-Hard. This means that it is very hard to obtain an optimal solution. Especially when the graphs becoming bigger; solving will take a lot of time and as the graph becomes too big it is not even possible anymore to obtain a solution in a reasonable time. As a result a lot of lower bound algorithms and heuristics have been developed.
The heuristics are used to obtain feasible solutions for the CARP. The outcomes of the lower bound algorithms are often used to evaluate the quality of heuristics. A selection of lower bound algorithms and heuristics is described in Chapter 3.

The feature of the CARP model is that it can be transformed into the Vehicle Route Problem (VRP), see Section 2.3.2., and vice versa. Transforming the CARP into a VRP makes it possible to use heuristics and algorithms of the VRP to solve the CARP. In paper [5] it is described how the transformation from a CARP into a VRP can be done. Note that after the transformation the problem became a much bigger and complete graph. The planar structure of the CARP graph is lost [8].

2.2.5. Capacitated Chinese Postman Problem
The Capacitated Chinese Postman Problem (CCPP) is a special case of the CARP. In the CARP just a subset \( R \) of the edge set \( E \) needs to be visited, in the CCPP all the edges in \( E \) have a positive demand, \( d_{ij} > 0 \) [15]. This means that all edges in the graph need to be visited at least once. The only extra constraint compared with the CPP is that there is a limitation on the capacity of the traveller. It means that, similar to the CARP, several circuits need to be obtained to ensure all the edges are traversed and the capacity constraint is not exceeded per circuit. Note that if the capacity of the traveller is bigger than the total demand of the graph the capacity is not a constraint and the problem becomes a CPP.

2.3. Node Routing Problems
The Node Routing Problem (NRP) is the general class of problems that involve all route planning problems on the nodes in a graph. The most known variant is the Traveling Salesman Problem; a salesman has to visit several cities and he wants the have the shortest possible route among all the cities. The aim for this kind of problems is to obtain a route with a minimal length whereby all the nodes in the graph are visited once. For all NRP it is required to visit all the nodes in the graph. The NRP problems do not have conditions on subsets like the ARP. That is because nodes that do not require a visited can be omitted from the graph; just the edges will be in the graph. A selection of variants of the NRP is described in the remainder of this section.

2.3.1. Hamilton circuit
The Hamilton circuit gets his name from the mathematician Hamilton [9]. The Hamilton circuit is the counterpart of the Euler circuit. Euler was wondering if it was possible to travel over each edge exactly one time and end at the starting point. Hamilton was wondering if it was possible to travel through every point in the graph exactly once and end in the starting point. If the routes are not ending at the starting point it is called an Euler path, respectively Hamilton path.

Determining if there exists an Euler circuit in a graph can be obtained in polynomial time, as described in Section 2.2.1. But for the Hamilton circuit it is much more complicated; at the moment no conditions have been found that ensure the presence of a Hamilton circuit. The mathematician Ore [9] discovered a characteristic that ensures the presence of a Hamilton circuit. It is a sufficient condition, but not a necessary one. The characteristic of Ore is as follow: For a graph \( G(V,E) \), if
\[ |E| \geq 3 \text{ and } \text{degree}(v) + \text{degree}(w) \geq |E|, \ v \in V, w \in V, \ for \ all \ (v,w) \notin E, \ \text{then there is a Hamilton circuit.} \]

Because there is not a set of conditions that ensures the presence of a Hamilton circuit in a graph it is difficult to determine a Hamilton circuit. Finding a Hamilton circuit is the only way that ensures the presence of a circuit.

**Determining if a graph contains a Hamilton circuit is NP-complete.**

The above is described in the paper of Laporte [19]. Laporte indicates that especially for big graphs it is very difficult and time consuming to determine a Hamilton circuit.

### 2.3.2. Traveling Salesman Problem

The Traveling Salesman Problem (TSP) is one of the most common studied combinatorial optimization problems [12]. Because the problem seems so easy but solving it is very difficult it is of big interest by scientists. The goal is to obtain an optimal route that traverses at least every node in the graph once. The TSP is the node routing equivalent of the arc routing problem CPP. The CPP tries to find a route that visits at least all the edges one time, the TSP wants so visit all nodes at least once. Both problems have the same goal; minimizing the total travelling length. The difference between both problems is the complexity in solving the problem to optimality.

**The TSP is NP-hard to solve.**

In the paper of Laporte [19] it is show that TSP is NP-hard. The explanation is based on a reduction of the TSP to a Hamilton circuit. There exist some special cases of the TSP that can be solved in polynomial time, for more information see [19]. Excluding this couple of special cases, the TSP is NP-hard to solve, what means that it is not possible to obtain the optimal solution in polynomial time. Especially when the graph becomes bigger it becomes time consuming to obtain the solution, due to the huge number of possible subsets [19]. Because of the big interest and the wide range of research in the TSP there is a huge amount of algorithms and heuristics to obtain a tour length. Some easy to use heuristics for the TSP can be found in [10].
2.3.3. Vehicle Route Problem

The Vehicle Route Problem (VRP) is described for the first time by Dantzig & Ramser in 1959 [6]. The VRP deals with the same kind of problem as the TSP. The only difference is that in the VRP there are $n$ travellers (or vehicles as the name suggests) instead of one traveller. The aim is to obtain at most $n$ tours that have together a total minimal length [20].

Different from the TSP, which exists of one main problem, the VRP exists of two problems. The first question is to which traveller a node should be assigned to and the second question is in which order the traveller should visit the nodes.

The VRP is a more complex model of the TSP, so this problem is NP-hard as well.

In Section 2.2.4 is already described that it is possible to transform the CARP problem into the VSP. This is a nice feature, because there are many heuristics and lower bound methods developed for the VSP. Thus by transforming the CARP into a VRP the CARP can also use all the heuristics and lower bound methods to obtain possible solutions. But this also holds the other way around. It is possible to transform the VRP into a CARP problem [8]. Then the VRP can be solved with algorithms and heuristics that are described for the CARP.

2.3.4. Capacitated Vehicle Routing Problem

The Capacitated Vehicle Routing Problem (CVRP) is a constrained variant of the VRP. The constraint is the total capacity a vehicle can handle. Whereas in the CARP the demand $d_{ij}$ is assigned to the edges, in the case of the CVRP the demand is assigned to the vertices of the graph. In the tour through all the vertices the total demand of the vertices should not exceed the capacity of the vehicle. Within this type there are several variants possible. For more information about the CVRP see the paper of Ralphs, Kopman, Pulleyblank & Trotter [21].

2.4. Models for mail delivery problem

There are two interpretations possible for the mail delivery problem. The first idea is that the mail delivery problem can be modelled as a variant of the Traveling Salesman Problem (TSP) with capacity constraints. That will be a CVRP where every vehicle represents one circuit of the postman. In this case all the mailboxes are represented by a vertex in the graph and every edge represents the distances between two mailboxes. Between every neighbour couple of nodes in the graph there is an edge. Note that the number of neighbour nodes differs per node due to the shape of the neighbourhood. The edges are undirected because walking from one mailbox to another is possible in both ways and the length does not depend on the walking direction.
From another point of view it seems like the mail delivery problem can be modelled as a CARP. In this case the nodes in the graph are representing the intersections of the streets in the neighbourhood. The edges are representing one side of a street. Then the demands of all mailboxes on one side of the street together represent the demand of the edge. The graph is undirected as walking by a side of a street can be done in both directions.

Both models give a good representation of the mail delivery problem but the CVRP graph is much bigger than the CARP graph. Assume a neighbourhood with 9 streets and 6 intersections and in total 220 mailboxes. Representing the situation as a CVRP means that there are 220 nodes and assuming that every node is connected in average with three other nodes, then there are 330 edges in the graph. That is a big graph.

Representing the neighbourhood with the 9 streets, 6 intersections and 220 mailboxes as a CARP means the graph will have 9 edges and 6 vertices. This is a relatively small graph compared with the CVRP graph. Because both models CVRP and CARP are NP-hard to solve, see Sections 2.2.4 and 2.3.4, it is advisable to use a small graph to solve the problem. That makes the problem not unnecessarily complex. So in this paper is chosen to represent the mail delivery problem as a CARP (or CCPP when there are no streets with a zero demand in the model).
3. Algorithms for mail delivery problem

In Chapter 2 an overview of models for route optimization problems is given. From this overview we conclude that the CARP model represents the mail delivery problem most appropriate. Because the CARP is an NP-hard problem it is difficult to obtain the minimal route length. For the CPP it is easy to obtain an optimal solution by the use of an integer linear programming model. Adding the capacity constraints to the model to obtain the minimal route length for the CARP makes the model very complex. That comes because of the conditions on all possible subsets that grow exponentially as the size of the graph increases, the conditions will be described in more detail in Section 3.2. In this chapter it is shown how the capacity constraint influences the complexity of the problem. This is shown by introducing in Section 3.1 the integer linear program model for the CPP and introducing in Section 3.2 the integer linear program model for the CARP. In the last section of this chapter an overview of other methods to handle the CARP are given.

3.1. Integer linear program for CPP

Edmonds & Johnson describe a linear programming algorithm for the optimal solution for the CPP in their paper of 1973 [7, p89]. This algorithm exists of two parts. The first part is based on extending the graph to obtain an Euler graph as described in Section 2.2.1. In the second part the exact tour can be found with the Fleury algorithm [9]. In this paper the focus is on finding the optimal minimal route length, so only the first part of the algorithm is described in detail. The second part, obtaining the actual route, is out of the scope of this paper and therefore not described here.

This first part of the method to obtain the optimal solution for the CPP is based on extending the graph with edges to obtain an Euler graph. This part can be represented as an integer linear programming problem as described in the paper [7, p89] of Edmonds & Johnson. The aim is to add extra edges to the graph so that every vertex has an even degree. An even degree of every node is the condition to have an Euler tour in a graph, see Section 2.2.1. The goal is to obtain a minimal set of extra edges that transforms the initial graph into an Euler graph. To understand the integer linear programming model, some definitions about graph $G(V, E)$ need to be introduced.

For the graph $G(V, E)$ we have the following notation:

\[
\begin{align*}
V & \text{ is the set of vertices} \\
E & \text{ is the set of undirected edges} \\
c_e & \text{ cost to traverse edge } e, \quad e \in E \\
a_{ne} & = \begin{cases} 
1 & \text{if edge } e \text{ is connected with node } n \\
0 & \text{otherwise} 
\end{cases}, \quad e \in E, n \in V \\
b_n & \text{ the degree of edges connected to vertex } n, \quad n \in V
\end{align*}
\]
To obtain an integer linear programming model some variables need to be defined. For the described graph above the following proper variables are defined.

\[
\begin{align*}
  x_e &= \text{the number of extra times edge } e \text{ is traversed} \quad e \in E \\
  w_n &= \sum_{e \in E} x_e \cdot a_{ne} \quad e \in E, n \in V \\
  z_n &= \text{an auxiliary variable} \quad n \in V
\end{align*}
\]

Now is it possible to introduce the goal of the integer linear programming in terms of the graph and the variables. The goal is:

\[
\text{minimize } \sum_{e \in E} c_e (x_e + 1)
\]

Taking the following constraints into account:

\[
\begin{align*}
  x_e &\geq 0 \text{ and integer} \quad \forall e \in E \\
  z_n &\geq 0 \text{ and integer} \quad \forall n \in V \\
  b_n + w_n &= 2z_n \quad \forall n \in V
\end{align*}
\]

The outcome of this integer linear programming model [7] is the minimal total route length that needs to be traversed to be able to serve every edge of \( E \). The goal of the model is to minimize the extra traversed edges with the condition that every edge in \( E \) is traversed at least one time so it can be served.

The first two constraints don’t need extra explanation. The third constraint ensures that every vertex in the extended graph will have an even degree. In this constraint \( b_n \) represents the degree of edges in the initial graph, and \( w_n \) represents the degree of the extra edges that are traversed in the extended graph. This sum must be even which is ensured by the equality with two times the auxiliary variable \( z_n \), where the latter is an integer. This integer linear programming model has in total \(|E| + 2|V|\) constraints.

### 3.2. Integer linear program for CARP

There are two integer linear programming models formulated for the undirected CARP [15]. The first is formulated by Golden & Wong [15] and the second is formulated by Belenguer & Benavent [15, 16]. Both models obtain a total minimal route length for the CARP. Only the method of Belenguer & Belenguer is described in this paper, because the two models are similar in complexity and just use another approach.

For the integer linear programming model for the CARP some assumptions have to make first. The first assumption is that the first node in the graph represents the depot (in the mail delivery problem this is the location where the bicycle is parked). The second assumption is that the number of circuits in the route is set in advance. When determining the number of circuits it should be taken into account that the number of circuits can meet the total demand of the graph. The third assumption is that every edge will be served with one visit.
While keeping the assumptions in mind, the definitions for graph $G(V, E)$ can be introduced.

$V$ is the set of vertices

$E$ is the set of undirected edge

$q_e$ demand of service for edge $e$ $e \in E$

c$_e$ cost to traverse edge $e$ $e \in E$

$R$ is the subset of all edges with $q_e > 0$ $R \subseteq E$

$m$ the predefined number of circuits

$k$ indicates which of the $m$ rounds $k = 1, ..., m$

$W_k$ the maximum capacity for round $k$

$S \subseteq V \backslash \{1\}$ subset of vertex set $V$ minus depot

$E(S)$ the set of edges that is with one side connected to a node in $S$ and with one side to a node outside of $S$

$E^+(S)$ the subset of edges from $E(S)$ with $d_e > 0$

In figure 3.1 a graph is displayed to make the last three definitions that are introduced above more clear. The green circle defines the subset $S$ and the red edges represent the edges with a positive demand. Note that subset $S$ is one example of a subset for the displayed graph. In this figure, $S$ is defined by the nodes $\{4, 5, 6, 7\}$, from which it follows that $E(S) = \{D, F, H, L\}$ and $E^+(S) = \{D, H, L\}$.

![Figure 3.1: A graph with subset $S = \{4, 5, 6, 7\}$](image-url)
To model the graph for the CARP as an integer linear programming model, some variables need to be introduced:

\[
x_{ek} \quad \text{the number of times edge } e \text{ is traversed in round } k \text{ (not served)}
\]
\[
y_{ek} \quad \begin{cases} 
1 & \text{if edge } e \text{ is served in round } k \\
0 & \text{otherwise}
\end{cases} \quad e \in E, k = 1, \ldots, m
\]
\[
y_{sek} \quad \begin{cases} 
1 & \text{if edge } e \text{ is with both sides in } S \text{ and } e \text{ is served in round } k \\
0 & \text{otherwise}
\end{cases} \quad e \in E, S \subseteq V \setminus \{1\}, k = 1, \ldots, m
\]
\[
z^s_k \quad \text{an auxiliary variable} \quad S \subseteq V \setminus \{1\}, k = 1, \ldots, m
\]

With the above definitions for the graph and the variables it is possible to determine the goal of the integer linear problem in terms of the definitions and variables. The goal is:

\[
\text{minimize } \sum_{e \in E, k} c_e(x_{ek} + y_{ek})
\]

Taken the following constraints into account:

\[
x_{ek} \geq 0 \text{ and integer} \quad \forall e \in E, k = 1, \ldots, m
\]
\[
y_{ek} = 0 \text{ or } 1 \quad \forall e \in E, k = 1, \ldots, m
\]
\[
z^s_k \geq 0 \text{ and integer} \quad \forall S \subseteq V \setminus \{1\}, k = 1, \ldots, m
\]
\[
\sum_{k=1}^{m} y_{ek} = 1 \quad \forall e \in R
\]
\[
\sum_{e \in E} y_{ek} \leq W_k \quad k = 1, \ldots, m
\]
\[
\sum_{e \in E(S)} x_{ek} + \sum_{e \in E^*(S)} y_{ek} \geq 2y_{Sfk} \quad \forall \text{edges } f \text{ that are total in } S, \forall S \neq \emptyset, k = 1, \ldots, m
\]
\[
\sum_{e \in E(S)} x_{ek} + \sum_{e \in E^*(S)} y_{ek} = 2z^s_k \quad \forall S \neq \emptyset, k = 1, \ldots, m
\]

The outcome of the model is the minimal total length of all the circuits to serve all edges with a positive demand. This is the length of all edges that are served and all edges that are just traversed (deadheading).

The first three constraints are evident. The fourth constraint ensures that every edge with a positive demand is served exactly once. The fifth constraint ensures that every round does not exceed the total capacity.

The sixth constraint ensures that every tour is connected to the depot, it disclose illegal tours. That means that if in a subset \( S \) an edge \( e \) is served in round \( k \), the round needs to go at least one time into \( S \) and at least one time out of \( S \). Constraint six checks this for every variable \( y_{Sfk} \), this variable
is 1 if edge $f$ is fully inside subset $S$ and is served in round $k$. It follows that $y_{Sfkr}$ is defined for every edge with $q_f > 0$ (is subset $R$), all subsets $S$ and all $m$ rounds. It means that there are $|R| 	imes |S| 	imes m$ constraints. In here is $|R|$ the number of edges in $R$ and $|S|$ is the number of subsets.

The seventh constraint ensures that the obtained routes are rounds. As described in Section 2.2.1, to obtain a circuit the route has to go out of a vertex as many times as it goes into the vertex. In this case a route has to go out of a subset as many times as it goes into a subset. Constraint seven checks for every subset and all the rounds if the property is met that an even number of edges goes into/out of the subset. To do the check, the auxiliary variable $z^S_r$ is used. The auxiliary variable is defined for every subset $S$ and for every round $m$. It means that for every subset $S$ there are $m$ constraints. So in total for this condition there are $|S| 	imes m$ constraints.

Because constraints six and seven have to hold for all the subsets $S$, the integer linear programming has many restrictions and variables. By just adding one extra vertex to the graph there arise a whole group of new subsets with each of the subsets two times $m$ constraints. By adding one vertex to a graph the number of constraints grows exponential. The number of subsets in a graph is $2^{|V| - 1}$, that explained the exponential growth.

In addition to the oversized growing of the number of constraints, the model is limited due to the assumptions that are made. The model assumes that the depot is at a fixed vertex 1. But for the mail delivery problem the depot location is not part of the input, the bicycle can be placed at any intersection. It means that the algorithm has to be repeated $|V|$ times to obtain from which vertex the absolute shortest total route can be obtained, and then still it is assumed that the bike will not be moved during the route. Another limitation of the model is that the number of circuits is part of the input, but for the postman it is not predefined how many circuits he should walk. The outcome ensures an optimal outcome for the given number of round. It is possible that with another number of rounds a shorter route could be obtained.

### 3.3. Other methods for the CARP

In Section 3.2 it becomes clear that it is complex (or even impossible as the graph becomes large) to obtain the optimal route length for the CARP. When it becomes impossible from any practical perspective to obtain the optimal solution other goals need to be formulated. Depending on the setting of the problem a feasible solution or a lower bound for the minimal total route length could be an alternative aim of the CARP problem.

In Section 1.2. is described that besides the mail delivery problem many other situations are dealing with a CARP. It has resulted in many studies of the CARP. Depending on the alternative aim of the problem many different approaches have been examined to obtain a feasible solution or a lower bound [8].

#### 3.3.1. Feasible solutions

Many heuristics have been developed to obtain feasible solutions for the CARP. The aim of these heuristics is not to minimize the total length of the route but to obtain several rounds that can be
used in practice while taking into account that the total route length should not be too long. The aim is to obtain a set of rounds with a reasonable total route length. Many heuristics have been developed to obtain feasible solutions [22, page 40]. They can be divided in three categories: simple constructive methods, two-phase constructive methods, and metaheuristics [15]. In the rest of this section a small selection of heuristics will be described to give background about how heuristics obtain feasible solutions.

The **construct-strike algorithm** belongs to the simple constructive heuristics. It is one of the first heuristics for the CARP described by Christofides in 1973 [22, page 40]. In the algorithm the rounds are constructed one by one. A feasible round is constructed and then removed from the graph. During the construction of the round, it is taken into account that the capacity cannot be exceeded and that, when the round is removed from the graph, the remaining graph does not consists of two (or more) disconnected sub graphs. The construction of rounds is repeated till there are no possible rounds left. Then for the leftover graph an Euler tour is constructed. The algorithm runs in $O(|E||V|^3)$ [22].

The **augment-merge algorithm** belongs to the simple constructive heuristics as well and is described by Pearn in 1991 [15]. The main idea of this algorithm is to merge routes. The first step is to obtain for all the edges that need service the shortest route. In the beginning every edge has its own route to the depot. The second step is to merge routes, starting from the longest route. Because there are some edges on the route that needs service as well and the capacity of the round is not exceeded the edge is merged into the route. The merging is repeated till merging is not possible anymore. The algorithm runs in $O(|V|^3)$ [15].

The **cluster-first, route second algorithm** belongs to the two-phase heuristics [15]. The algorithm exists of two parts. In the first part, all the edges are divided in $k$ clusters with a total demand that is not exceeding the capacity. That can be done with several algorithms, for example with a greedy algorithm. In the second part routes for each cluster are constructed.

The metaheuristics are higher level algorithms to obtain feasible solutions. Sometimes these algorithms are based on improving a constructive algorithm, or they combine some constructive algorithms. Often the metaheuristics are developed for a special case of a CARP. Due to the complexity of the metaheuristics, these are not described in this paper. More information can be found in [8, 15, 22].

### 3.3.2. Lower bounds

The lower bound methods have the aim to obtain a lower bound for the total route length for the CARP. The feasible methods search for a feasible solution; the lower bound methods just derive a minimal length for the total route. The lower bound does not guarantee that a feasible solution exists for that obtained route length. It only ensures that it is not possible to obtain a feasible solution with a smaller total route length. Often the lower bound methods are used to determine how good the feasible algorithms perform.

Relaxations of the integer linear programming model described in Section 3.2 are obvious lower bounds for the CARP. But, next to these relaxations, other methods for lower bounds are described
as well. The first lower bound method is described by Golden and Wong in 1981, the so called matching lower bound [8]. The method is based on a graph $H = (U, B)$ where $U$ is the union of three disjoint sets based on the initial graph and $B$ is the set of edges between these sets. How the algorithm works exactly can be found in [15]. After Golden and Wong many other lower bounds methods have been described. For an overview of the different lower bound methods, see [8, 15].

When more conditions are integrated in the lower bound method, the method will obtain an outcome that is closer to the optimal solution. But adding conditions makes the method more and more complex whereby the running time increases. In the situations where the integer linear programming model becomes too time consuming to solve, the lower bound method is still capable to obtain a solution.

### 3.3.3. CPP and CARP compared

In the previous sections of this chapter it became clear that obtaining an optimal solution for the CPP could be done easily with an integer linear programming model. But expanding the model with capacity constraints makes the model much more complex. For the integer linear programming of the CPP there are $|E| + 2|V|$ constraints involved, for the CARP there are $2(|E| * m) + 2(|S| * m) + |R| * |S| * m + m$ constraints involved. Note that the number of subsets $S$; $|S| = 2^{V-1}$.

| $|V|$ | $|E|$ | $|R|$ | $m$ | CPP | CARP |
|------|------|------|-----|-----|------|
| 3    | 4    | 4    | 2   | 10  | 66   |
| 4    | 7    | 7    | 2   | 15  | 174  |
| 6    | 10   | 8    | 2   | 22  | 682  |
| 6    | 10   | 10   | 2   | 22  | 810  |
| 10   | 15   | 12   | 3   | 35  | 21,597 |
| 10   | 15   | 15   | 3   | 35  | 26,205 |
| 10   | 15   | 15   | 4   | 35  | 34,940 |
| 15   | 25   | 18   | 4   | 55  | 1,310,924 |
| 15   | 25   | 19   | 4   | 55  | 1,376,460 |
| 20   | 25   | 20   | 4   | 65  | 46,137,548 |
| 20   | 25   | 25   | 4   | 65  | 56,623,308 |
| 50   | 80   | 75   | 20  | 180 | 866,942,928,268,824,000 |

**Table 3.1: Number of constraints for integer linear programming of CPP and CARP.**

In Table 3.1 is shown how many constraints are needed for some instances of a CPP and CARP integer linear programming model. In the table can be seen that the number of constraints for the CARP is growing much faster than for the CPP. The first four instances are a bit too small to represent a real mail delivery problem, but already for these graphs the number of constraints is very high for the CARP. The next seven instances are realistic representations for the mail delivery problem. In these instances the numbers of constraints are for the CPP still reasonable, but for the CARP there are thousands of constraints that need to be fulfilled. These are so many requirements that it is not easy to implement an instance of CARP as an integer linear programming model. In addition, when it would be implemented it will take an unreasonable time to obtain a solution.
4. Conclusions and further work

In this chapter the conclusions and some suggestions for further work for this research will be given.

4.1. Conclusions
The general conclusion of the research is that it is very complex to obtain the minimal total route length for the mail delivery problem. The conclusion is constructed from two sub conclusions.

In the first part of the research is investigated how the mail delivery problem could be modelled. That resulted in an overview of several route optimization models that are described in the literature. It became clear that the Capacitated Arc Routing Problems is the best model to use to describe the mail delivery problem. It is also possible to model the mail delivery problem as a Capacitated Vehicle Routing Problems but this makes the problem unnecessary complex.

In the second part of the research, integer linear programming models for the Chinese Postman Problems and the Capacitated Arc Routing Problems are reviewed. That resulted in the second conclusion: it is very hard to obtain the optimal total route length for the Capacitated Arc Routing Problems. For the Chinese Postman problems it is relative easy to obtain a solution, but by adding the capacity (or any other) constraint to the model, it becomes much complex to solve the model because of the extra constraints. These extra constraints make that solving the problems is NP-hard, because the number of constraints grows exponential. Due to this complex solving model, there is done a lot of research to alternative methods to obtain feasible solutions and lower bounds.

The integer linear programming model for CPP is applied to the example of Chapter 1. In figure 4.1 the lengths of the edges are given. The outcome of the integer linear programming for the CPP gives that just one street should be traversed double to extend the graph representation into an Euler graph. In figure 4.2 the outcome is illustrated with the extra traversed street side.

The example above as a CARP results in an integer linear programming model with a lot of constraints. In this example are 4 nodes and 7 edges. Assume that the postman can deliver all the mail in three circuits and the bike is not moved during the delivery. That results in an integer linear programming mode with \[ 2(|E| * m) + 2(|S| * m) + |R| * |S| * m + m = 2(7 * 3) + 2(64 * 3) + 7 * 64 * 3 + 3 = 1759 \] constraints. This has to be solved for every node, thus 4 times to obtain the absolute best solution. This is too big to implement and solve in a reasonable time.
4.2. Further work

The main focus in this paper is to find a model that fits the capacity constraint mail delivery problem and determine how difficult it is to solve this problem. Because it is very complex to derive the optimal route length no implementation is done. It is suggested to implement some instances of mail delivery problems to see the actual influence of the capacity constraints on the total route length. When the instances are too big to solve the CARP with the integer linear programming model it is suggested to implement other methods; feasible heuristics and lower bounds approaches in order to still see the influence of the capacity constraint in the total route length.
5. Bibliography


[22] D. Ahr, Contributions to Multiple Postmen Problems, Ruprecht-Karls-Universitat, 2004

[23] Postman at Sandd: _confidential_, Sept 1, 2013