Chapter 2

State of the Art

2.1 A Historical Perspective

The study of arc routing problems began on August 26, 1735 when Leonhard Euler presented his solution to the Königsberg bridge problem, Sachs et al. 1988 [127]. The Königsberg bridge problem was stated as follows

Problem 1 (The Königsberg Problem). *Given the city of Königsberg with its seven bridges, is it possible to go for a walk, starting and ending the same place and passing each of the bridges exactly once?*

Being a mathematician, Euler formulated the problem, as a graph theoretic problem which he then solved in the general case. This problem, known as the Euler tour problem is formulated as

Problem 2 (The Euler Tour Problem). *Given a connected graph $G = (N, E)$ find a tour that visits every edge in E exactly once, or determine that no such tour exists.*

Euler proved that an Euler Tour exists if and only if every node in G has even degree. Later Fleury presented an algorithm for constructing an Euler Tour, Fleury 1883 [70].

The next arc routing problem to be studied was the Chinese Postman Problem first suggested by the Chinese mathematician Kwan Mei-Ko 1962 [114]. The problem is formally stated as

Problem 3 (The Chinese Postman Problem (CPP)). *Given a connected graph G = (N,E,C) with distances on the edges, find a tour, which passes through every edge at least once and does this in the shortest possible way.*

When the underlying graph is completely directed or completely undirected, the Chinese Postman Problem can be solved in polynomial time, see Edmonds and Johnson 1973 [61] and Christofides 1973 [37]. However, when the underlying graph is mixed, the problem becomes $\mathcal{NP}$-hard, Papadimitriou 1976 [120]. Many variants of the problem have been studied, including the Windy Postman Problem, Minieka 1979 [116] and the Hierarchical Postman Problem, Dror et al. 1987 [60]. For a survey on the Chinese postman problem and some of its variants we refer the reader to Eiselt et al. 1994 [67].

The next arc routing problem to be considered was the Rural Postman Problem, which was first suggested by Orloff 1974 [119]. The problem is formally stated as

Problem 4 (The Rural Postman Problem (RPP)). *Given an undirected graph* $G = (N, E, C)$, *where* C *is the cost matrix for the edges, find a minimum cost tour, which passes through every edge in a subset* $R \subseteq E$ *at least once.*

The Rural Postman Problem was proved to be $\mathcal{NP}$-hard by Lenstra and Kan 1976 [101]. The difference between the Chineese Postman Problem (Problem 3) and the Rural Postman Problem (Problem 4), is that in the latter only a subset of the edges need to be traversed. Jansen 1992 [96] gave a 3/2-approximation algorithm for the problem and notes that the problem can be solved to optimality in polynomial time if the graph spanned by the set R consists of only a fixed number of components. The hardness of the problem is determining how the tour should connect the various components. Several heuristic procedures have been suggested for RPP, for example Pearn and Wu 1995 [124], Hertz et al. 1999 [89] and Córdoba et al. 1998 [54]. Procedures to solve the problem to optimality have been suggested by Corberán and Sanchis 1994 [45] and Ghiani and Laporte 2000 [79]. It can be shown that the class of RPP and the class of Traveling Salesman Problems (TSP) are equivalent. See for example Wøhlk 2002 [94] for a proof. For a more complete survey of the Rural Postman Problem we refer the reader to Eiselt 1995 [68].

Along with the RPP, Orloff 1974 [119] also suggested another $\mathcal{NP}$-hard problem, called the General Routing Problem, which is a combination of arc routing and node routing. Formally this problem is stated as

Problem 5 (The General Routing Problem (GRP)). *Given an undirected graph* $G = (N, E, C)$, *where* C *is the cost matrix for the edges, find a minimum cost tour, which passes through every edge in a subset* $R_E \subseteq E$ *and visits every node in a subset* $R_N \subseteq N$ *at least once.*

Many variations of the problems have also been considered. For example all the problems have been studied where the underlying graph is directed, undirected or mixed, and some of the problems have been considered with a different objective function. We would just like to mention one variation of the RPP, which we find interesting, namely the Rural Postman Problem with Deadline Classes suggested by Letchford and Eglese 1998 [64]. Here the set R of required edges are partitioned into several sets, which in turns are ordered so that all the edges in an earlier set must be traversed before any of the edges in a later set.

The final arc routing problem wes suggested by Golden and Wong 1981 [81]. The Capacitated Arc Routing Problem is formulated as

Problem 6 (The Capacitated Arc Routing Problem (CARP)). *Given a connected undirected graph* $G = (N, E, C, Q)$, *where* C *is a cost matrix and* Q *is a demand matrix, and given a number of identical vehicles each with capacity* W *(where* $W \geq \max q_{ij}$*), find a number of tours such that 1) Each arc with positive demand is serviced by exactly one vehicle, 2) The sum of demand of those arcs serviced by each vehicle does not exceed* W*, and 3) The total cost of the tours is minimized.*

The Capacitated Chinese Postman Problem (CCPP) is the variation of the CARP where every edge in the graph has a strictly positive demand. This problem was first considered by Christofides in 1973 [37], as a generalization of the CPP. Both the CARP and the CCP are $\mathcal{NP}$-hard by reduction from the Partitioning Problem as proved by Golden and Wong 1981 [81] who also showed that obtaining a solution within a factor 3/2 of the optimal is $\mathcal{NP}$-hard, even when the triangle inequality is respected. In the same paper the authors showed a way of transforming the Vehicle Routing Problem (VRP), which is a node routing problem, into the CARP. Thereby the VRP can be considered as a special case of the CARP. Assad et al. 1987 [11] showed a method for transforming the CARP into the VRP, making the CARP a special case of the VRP. Thereby the two classes of problems become equivalent. From this it also follows that obtaining a solution to the CARP within a fixed approximation factor is a $\mathcal{NP}$-hard problem, if the triangle inequality is not respected, since the Travelling Salaman Problem (TSP) is a special case of the VRP. Reasently both Baldacci and Maniezzo 2004 [14] and Longo et al. 2004 [53] have presented tighter transformationd from the CARP to the VRP. These transformations allow for the use of VRP techniques to solve the CARP. For all three transformations, the resulting VRP instance requires either fixing of variables or the use of edges with infitite cost.[1]

We will use the remeinder of this chapter to give an overview of the literature on the CARP.

2.2 Solution of the CARP

Solution procedures for an optimization problem can be partitioned into four classes; Approximation Algorithms, Problem Specific Heuristics, Meta Heuristics, and Exact Algorithms. Approximation algorithms are designed specifically for one problem type, and are guaranteed to return a solution of cost at most some constant times the cost of an optimal solution to the problem. This is sometimes referred to as a performance quarantee. No approximation algorithm have been proposed for the CARP, but Jansen 1993 [97] gives an approximation algorithm for a capacitated version of the general routing problem, which contains the CARP as a special case. Problem Specific Heuristics are also algorithms designed directly for the problem at hand. Unlike approximation algorithms, heuristics are in general not guaranteed to give solutions within any quality bound, but are expected to perform well in practice. We will consider the existing problem specific heuristics for the CARP in Section 2.2.1. The second class of solution procedures, which we will consider in Section 2.2.2, are Meta Heuristics. These algorithms are constructed to work on any problem for which we can define a cost structure and a neighborhood. By defining the cost structure, i.e. the cost of a solution and a neighborhood, i.e. a way of getting from one solution to another, the meta heuristic can be applied to specific problems. Finally, we consider the class of Exact Algorithms in Section 2.2.3. These are algorithms that guarantee to find an optimal solution to the problem, but have exponential worst case running time.

[1]Formally a sufficiently large constant is used.

2.2.1 Problem Specific Heuristics for the CARP

When introducing the CCPP, Christofides 1973 [37] also suggested a heuristic called the **Construct-Strike Algorithm**. The algorithm repeatedly goes through two steps. First, tours are constructed such that 1) the demand on each tour does not exceed the vehicle capacity and 2) when removing the demand edges on the tour, the remaining graph is still connected. In the second step the remaining graph is made even and the degree of the depot made at least two by adding artificial edges of a minimum cost perfect matching between the odd nodes in the graph. When all demand edges are on a tour the algorithm stops. Even though the algorithm is constructed for the CCPP it can be used for the CARP by adding artificial edges to make the graph induced by demand edges connected.

The first heuristic for directly solving the CARP was the **Augment-Merge Algorithm**, first suggested by Golden and Wong in 1981 [81], and explained in detail in Baker et al. in 1983 [13]. The Algorithm consists of two phases. First, cycles are constructed with one demand edge on each. Second, if demand on a smaller cycle can be moved to a larger cycle, without changing the route for that cycle, the demand is moved and the smaller cycle is discarded. In the second phase cycles are merged if this can be done without violating capacity constraints and yields a solution of lower cost.

The second heuristic for solving the CARP was the **Path-Scanning Algorithm** suggested by Baker et al. in 1983 [13]. In the Path-Scanning Algorithm tours are constructed one by one by repeatedly selecting the next edge according to one of five criteria. When the vehicle capacity is exhausted, a shortest path route back to the Depot is used. A complete solution using each of the five criteria is constructed and the best of the five is chosen to be the final solution. In the five criteria the next edge, (i, j), is chosen so that 1) The distance, c_{ij}, per unit demand is minimized; 2) the distance, c_{ij}, per unit demand is maximized; 3) the distance from node j back to the depot is minimized; 4) the distance from node j back to the depot is maximized; 5) if the vehicle is less than half-full, the distance from j to the depot is maximized, otherwise this distance is minimized.

With the **Parallel-Insert Algorithm** by Chapleau et al. 1984 [36] the authors try to construct routes that are well balanced, besides from minimizing cost. First it is estimated how many tours are needed, and each of these tours is initialized with an edge far from the Depot. Now the tours are constructed in parallel using two steps, 1) Given an edge, determine which tour to insert it in to minimize the insertion cost, and 2) Given a tour, determine which of the remaining edges should be inserted in that tour. The algorithm iterates between the two steps as long as possible. Finally extra tours may need to be initialized to service all the edges.

In Pearn 1989 [122] the author presents modified versions of two already known heuristics, the Construct-Strike Algorithm and the Path-Scanning Algorithm. In the **Modified Construct-Strike Algorithm** the second requirement of the tour construction of the original algorithm is removed, which leads to the following algorithm. Tours are constructed by repeatedly choosing the edge that maximize the least quantity path back to the Depot. Now, after removing the demand edges of the constructed tours, if the remaining graph is disconnected, a Minimum Spanning Tree between the components is constructed before the graph is made even in the same way as in the original algorithm. In the **Modified Path-Scanning Algorithm**, instead of constructing a complete solution using just one of the five criteria, here we choose at random, which of the criteria to use at each step of the algorithm. This way a several solutions are constructed, and the best of these are chosen. This algorithm is sometimes referred to as the **Randomized Path-Scanning Algorithm**.

The last problem specific heuristic for the CARP is the **Augment-Insert Algorithm** suggested by Pearn 1991 [123]. Here, the demand edges are considered in decreasing order with respect to the distance to the depot. For each edge the shortest tour containing the edge is constructed and the tour is augmented with demand edges already on the tour, and the serviced edges are removed from the graph. This is repeated until all demand edges are disconnected from the depot. At this point all edges are restored and the process is repeated, except that now the augmentation may include edges that are not already on the tour. How expensive this insertion may be depends on a parameter, which in turn is changed over several runs of the algorithm, and the best solution is chosen.

2.2.2 Meta Heuristics for the CARP

In **Simulated Annealing** one repetedly picks any solution in the neighborhood and shifts to this solution with some probability p, if the solution is worse than the one at hand. If the solution is better we always shift. The probability p decreases during the process according to some prespecified cooling scheme. Eglese 1994 [63] presents a Simulated Annealing Algorithm for a winter gritting problem, which is modeled as a CARP with extra complicating constraints specific to the case studied.

In **Tabu Search** we have a list of tabu solutions (or a list of tabu changes), which is a list of solutions that we may not switch to (or a list of changes that we may not perform). This tabu list is dynamic and will often consist of the ones that we have recently obtained. Given a solution, we search the neighborhood to find the best solution that is not in the tabu list. This is continued a fixed number of times. Several Tabu Search Algorithms have been constructed for the CARP. First an algorithm called CARPET by Hertz et al. 2000 [90] was suggested. The next Tabu Search Algorithm by Amberg et al. 2000 [7] is made for the Multi Depot version of the problem. Recently Greistorfer 2003 [85] has combined Tabu Search with a so-called Scatter Search, which is a method to combine a number of solutions to a problem, to construct a **Tabu Scatter Search** for the CARP.

There are several variations of **Genetic Algorithms**, which are mainly variants over the following: We start with a set of solutions called a generation. From two solutions in the set, we construct two new solutions by crossover. This way we construct a whole new set of solutions. We add the two sets and let the best half be the new generation. We continue in this way a fixed number of times. Lacomme et al. 2001 [104] presents a Generic Algorithm for the CARP. Lacomme et al. 2004 [107] present a variation of a generic algorithm, which they refer to as a **Memetic Algorithm** for the CARP. Here the crossover is performed on a giant tour, which is split subsequently.

One of the younger generations of Meta Heuristics is that of an **Ant Colony System**. Here one tries to mimic the way ants lay down pheromone on the ground on their way to food resources depending on the distance to and the quality of the resource. Consequently, paths with large amounts of pheromone are more likely to be used by other ants. Using these ideas Doerner et al. 2003 [57] have constructed an Ant Colony Heuristic for the CARP.

In **Guided Local Search** the objective is improved in every iteration until a local optimum is reached. Buellens et al. 2003 [24] presented a Guided Local Search Algorithm for the CARP. In their algorithm the objective to be minimized is the dead-heading[2] distance, where the distance of each edge is penalized according to some function, which is adjusted throughout the algorithm.

2.2.3 Optimal Solution of the CARP

In some cases a near optimal solution of a problem is not satisfying. In that situation one must find ways to construct an optimal solution to the problem. Since the CARP is $\mathcal{NP}$-hard, we do not expect to be able to find an optimal solution to the problem in polynomial time. The procedures that are often used to reach this goal have exponential worst case running time but are often much faster than that in practice. Variations of two algorithms, Branch and Bound and Cutting Plane, are the most common when seeking optimal solutions. These are also the two that have been applied to the CARP.

Both strategies are based on Integer Linear Programming Formulations (ILP). An ILP of a problem is a formal mathematical description of the problem, where it is formulated with an objective function, which is to minimize or maximize some function over some decision variables, and a set of constraints that must be satisfied in any feasible solution to the problem. For ILPs at least a subset of the variables are required to have integer value. The first ILP formulation for the CARP was presented by Golden and Wong 1981 [81] and is based on directed variables even though the problem formulated is undirected. A formulation using undirected variables is presented by Belenguer and Benavent 1991 [20]. In his PhD. Dissertation, Letchford 1996 [110] gives several ILP formulations of the CARP, and derives additional valid inequalities and separation algorithms for the problem.

There are many variants of **Branch and Bound**. The idea is to build a decision tree where the solutions are enumerated, but to use lower bounds and solution values to cut off part of the tree and to decide which part of the tree is most promising and should be explored next. Hirabayashi et al. 1992 [93] constructed an algorithm for solving the CARP to optimality by using Branch and Bound ideas. The authors use their own Node Duplication Lower Bound (NDLB), Hirabayashi et al. 1992 [92], to calculate lower bounds for the subproblems and branch on a single edge of the node duplicated network. Using this algorithm, the authors are able to solve a set a CARP instances containing from 15 to 50 demand edges to optimality.

In **Cutting Plane** an LP relaxation of the problem is solved. If the resulting solution is not integral, one seeks to find a valid inequality that can cut off the current fractional solution without cutting off any integer solutions. This is repeated until the LP relaxation of the problem is integral. Again there are many variations of the algorithm, particularly in the construction of valid inequalities and in the choice of which inequalities to include. It is not always possible to reach an integer solution by cutting planes, in which case one ends up with a fractional lower bound to the problem. This amounts from two facts: First, not all valid

[2] Traversing an edge without servicing it is called dead-heading.

inequalities for the problems are known. Second, even in the cases where the recipe for all the valid inequalities are known, at least one class of the inequalities cannot be generated in polynomial time (unless $\mathcal{P} = \mathcal{NP}$). The second fact comes from the original problem being $\mathcal{NP}$-hard. If a great number of classes of valid inequalities are known for the problem, one can often in practice reach an optimal solution by this method. Belenguer and Benavent 2003 [22] presented a cutting plane algorithm for the CARP, which is partly based on some classes of valid inequalities presented by the same authors in Belenguer and Benavent 1998 [21]. Using their algorithm, the authors are able to reach the best existing lower bound for all test instances, and can improve the existing lower bounds for several instances. They reduce the average gab between upper and lower bound to be less than one percent for three sets of instances[3], each containing between 11 and 97 demand edges. For a harder set of test instances[4], containing between 51 and 140 demand edges, they reduce the average gab to 2.4 percent.

Lately, attempts have been made to combine the above algorithms to get faster solution methods. Combining Branch and Bound with Cutting Plane results in an algorithm called **Branch and Cut** where in each node of the search tree one tries to add valid inequalities to the problem a number of times before branching. Clearly there is a trade-off between adding cuts and branching since adding cuts results in a more complicated lower bound calculation which, on the other hand, does not need to be performed as many times. More exotic combinations of solution procedures are also possible, for example a combination of Branch and bound, Cutting Plane and Column Generation will result in a **Branch, Cut and Price algorithm.** To our knowledge none of these combinations have been applied to the CARP, but have proven successful for other combinatorial optimization problems such as the Vehicle Routing Problem (VRP).

Finally, it should be mentioned that also a directed version of the CARP has been considered with the goal of deriving optimal solutions. In his PhD. Dissertation, Welz 1994 [133] considered this problem, and presents valid inequalities and separation algorithms for his ILP formulation of this problem.

2.3 Lower Bounds for the CARP

A Lower Bound for a problem is a value, LB, for which it can be proved that any feasible solution to the problem must have a cost at least as large as LB. Lower Bounds have several purposes. First of all, they are used to measure the quality of a solution obtained by a heuristic solution method. As indicated in Figure 2.1, it is desirable to get a lower bound that is as high as possible. When a valid lower bound and the cost of a feasible solution coincide the lower bound automatically proves optimality of the solution obtained. Furthermore, Lower Bounds are used in Branch and Bound procedures where an optimality proof for the obtained solution is needed. Often the bound used here is an LP-relaxation lower bound, but any valid lower bound procedure can be used.

Several lower bounds have been constructed for the CARP, most of which are based on matchings. One could classify the bounds according to their type, e.g. whether they consist

[3]These sets are referred to as the Gdb instances, the Val instance, and the Kshs instances.

[4]These instances are referred to as the Eglese instances.

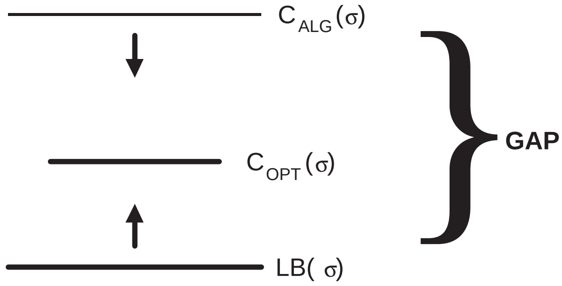

Figure 2.1: Gap between Valid Solution and Lower Bound

of one or several iterations. In Figure 2.2 the relationship among the various lower bounds for the CARP is shown. Here an arrow from bound x to y indicates that bound x outperforms bound y, i.e. $x(\sigma) \geq y(\sigma)$ for any instance σ of the problem. The Hierarchical Relaxations Lower Bound (HRLB) has only been experimentally compared with the other bounds, so even though it performs well on the tested instances it has not been proved to outperform any of the other bounds for all instances, which is why there is no relationship arrows to and from that bound.

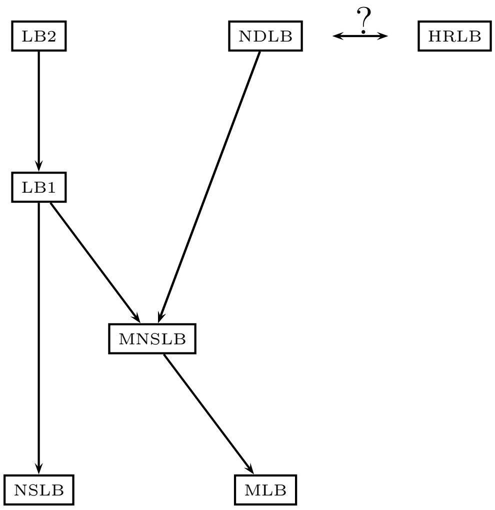

Figure 2.2: Relationships among Lower Bounds

The **Matching Lower Bound (MLB)** was presented in the original paper on the CARP by Golden and Wong 1981 [81]. It amounts to estimating the number of vehicles needed to service the graph and calculating the cost of a complete perfect matching where we ensure that at least the required number of tours start in the Depot.

The **Node Scanning Lower Bound (NSLB)** was presented by Assad et al. 1987 [10]. NSLB is based on logic arguments that bound the length of the path the vehicles must traverse before servicing the first demand edge on a tour, and after servicing the last one.

The **Matching - Node Scanning Lower Bound (MNSLB)** by Pearn 1988 [121] is a mix between MLB and NSLB. In this bound one minimizes over all possible combinations of the two algorithms by first considering the paths in the beginning and end of the tours for some number of tours, and then constructing the MLB matching for the remaining tours. This is done for all combinations of the required number of tours.

Benavent et al. 1992 [23] suggested ways of improving MLB. Their first algorithm, **LB1** uses the same idea as the MNSLB but instead of iterating over several combinations as in that algorithm, the LB1 tries to combine everything in one iteration in the following way: A graph is constructed with several copies of the nodes close to the Depot but only one copy of the rest of the nodes. In this graph a Minimum Cost Perfect Matching is constructed, where the correct number of copies of the Depot ensures that the required number of vehicles are used. In the same paper the authors present two lower bounds, **LB3** and **LB4** where the number of vehicles used is fixed.

The four lower bounds, MLB, NSLB, MNSLB, and LB1 all estimate the number of artificial edges we need to have incident to the depot node. In other words: They consider the cut $(\{1\}, G \setminus \{1\})$. In his PhD Dissertation Zaw Win 1988 [135] suggested that one considers not only one cut as in the previous bounds, but a whole family of disjoint cuts. Doing this makes the complexity of the algorithms larger but gives stronger results. The next couple of Lower bounds for the CARP are based on Zaw Win's idea. For each cut, $(U, \overline{U})$ they give a lower bound on the cost of the edges in $G(U)$, where U includes the depot. This estimate will get larger as the set U grows. At the same time the idea is to estimate the cost of the edges in $G \setminus G(U)$, where $G(U)$ is the graph induced by U. Zaw Win does this by a construction similar to the one used in MLB. Benavent et at. 1992 [23] improved this result in their lower bound **LB2** by estimating the cost in $G \setminus G(U)$ using their own algorithm LB1. The LB2 lower bound is explained in details in Chapter 3.

Hirabayashi et al. 1992 [92] suggest another lower bound, the **Node Duplication Lower Bound (NDLB)**, which is again based on calculating a minimum cost perfect matching in a specially constructed graph. Here the graph is constructed by making two nodes for each demand edge, one for each end, and connecting these by using shortest path distances from the original graph, and then removing edges that would lead to illegal tours. The NDLB is explained in detail in Chapter 3.

The final lower bound that has been suggested for the CARP is the **Hierarchical Relaxation Lower Bound (HRLB)** by Amberg and Voß 2002 [8]. This is an iterative bound as is the LB2 but here the cuts that one iterates over are not disjoint. HRLB starts out by solving the CPP relaxation of the CARP. In each iteration more constraints are added to the problem, the cutset is extended and the relaxation is solved again.

In his PhD. dissertation, Ahr 2004 [2] reasently improved the lower bound which we present in this dissertation. The improvements are based on a journal varsion of our work.

2.4 Applications of the CARP

There are many applications of the CARP and its variations in both public and private business. Here we will describe some of the applications studied in the literature.

2.4.1 Street Sweeping

Bodin and Kursh 1978 [30] study the problem of routing street sweepers in the cities. This problem is clearly a variation of the CARP since it can be modeled as a graph where the links (edges and/or arcs) must be serviced. The objective here is to minimize the total travel cost. The application studied by the authors has the restriction that a street cannot be swept during parking hours. This results in a problem in which each link has a time window associated in which the service of that link must take place. We will refer to this problem as the Capacitated Arc Routing Problem with Time Windows (CARP-TW). Bodin and Kursh 1979 [29] give a detailed description of a computer system to solve the problem. Their algorithm starts by solving a transportation problem to decide which links must be traversed an extra time without being serviced.[5] Then, to take care of further restrictions such as U-turns that must be penalized, an assignment problem is solved, and finally an Euler tour is constructed. This algorithm is finally put in a larger framework where capacity constraints and time windows are taken care of. It may be noted, that the time windows are handled in a serial manner and are not included in the solution algorithm itself.

Street sweeping in rural areas are considered by Eglese and Murdock 1991 [66]. They argue that unlike in urban areas, there are very few one-way streets in rural areas. Also, because of the type of roads it is possible to perform U-turns in these environments, and because of lack of parking restrictions one does not have to worry about time windows on the streets. For these reasons, the routing of street sweepers become easier in rural areas. On the other hand, because the area is often larger there will be several tip sites and a good routing algorithm must take this into consideration. The authors note that because each road can be represented by two arcs pointing in opposite directions, a tour can always be constructed that services both sides of a street on the same tour. In a sense this is done by noting that a tour can be considered a tree on the street network. The tour can then be constructed by traveling along the tree while always choosing a left must turn if possible, and if not by making a U-turn. Modifications are then made to deal with dead-ends and the likes, and the tip site is chosen as the one nearest to the end of the tour.

2.4.2 Winter Gritting

Eglese and Li 1992 [65] consider the problem of spreading a de-icing agent on roads to prevent them from becoming dangerously slippery. This is often referred to as Winter Gritting and can be modeled as a CARP, since the problem is to service a set of streets in a network. The streets are serviced by Capacitated vehicles, where the capacity constraint is on the amount of de-icing material the truck can keep and the time the whole gritting takes. The objective of the problem is to service the streets with a minimum cost with respect to the constraints just mentioned. In contrast to the case of street sweeping, when considering winter gritting each street can be serviced in both directions at once with the exception of highways with multiple lanes in each direction. This means that the relative simple solution method cannot be applied to this situation. The authors do not go into details about how they choose to solve the problem. In stead they give an interesting discussion on how efficiency of the constructed

[5] This is often called dead-heading.

solutions can be measured, and how the type of network (rural versus urban area) influences this measure.

Eglese 1994 [63] also considered the routing of winter gritting vehicles. The author considers a setup where the streets are partitioned into a number of categories, where those in category 1 must be treated within two hours of call-out, those in category 2 must be treated within four hours of call-out etc. This means that the problem becomes time constrained. These time constraints could be handled as wide time windows or, as is the case in the paper, by considering the categories separately. As before this winter gritting problem is also constrained by the capacity of the vehicles and thereby by the street length they can service. In addition the problem here is a multiple depot problem, i.e. there are several possibilities as to where the vehicles can be stationed and loaded for gritting. The network is partitioned into elementary cycles at every node with a degree of more than two and a cycle-node network is then constructed containing a node for each of these cycles. Two nodes in the cycle-node network are connected by an edge if the corresponding cycles in the original network have a node in common. The location of the Depots are now fixed, and tours are constructed by making rooted trees in the cycle-node network. If necessary tours are merged. Finally the tours are used as a starting solution for a simulated annealing algorithm where the time and capacity constraints are relaxed and penalized in the objective function.

Cattrysse et al. 2002 [33] also consider the problem of spreading salt for winter gritting. The authors mainly consider a setup with multiple Depots and are interested in design of districts for the gritting operation. A district is a geographical area associated with a single Depot, and are constructed on a long-term planning level, whereas the routing of the vehicles within each district can be modified on a short term level. The authors assume that the location of the Depots are fixed and describes an algorithm to construct the districts in the following way. The network is first partitioned into elementary cycles which are in turn assigned to a depot as a whole. The assignment of cycles to Depots (and thereby to districts) is done such that cycles that are much closer to one depot than to any other depot are assigned to that depot, and so on, while making sure that the borders between districts are relatively clear. Finally, each district is considered in turn and the routing of the gritters are done using the Augment-Insert heuristic by Pearn 1991 [123].

2.4.3 Refuse Collection

Bodin et al. 1989 [27] analyzes the routing of sanitation vehicles. The core of that problem is clearly a CARP, since it requires service along the streets, and the problem is capacitated by the amount of refuse in each truck and by the number of working hours. Furthermore, the goal is to minimize the operating cost. The authors model consider, not only the routing part of the problem, but the whole aspect of collecting and storing data for easy access by the routing algorithms, the generation of the networks and the final reporting. The routing part of the problem is handled by constructing one giant tour using a similar algorithm as the one used by Bodin and Kursh 1978 [30]. The giant tour is then partitioned into a set of relatively equal sized tours respecting the capacity constraints.

Mansini and Speranza 1998 [113] consider the problem of refuse collection from a more tactical point of view. They focus on minimizing the maximal load of the workers when the collection of refuse is done a fixed number of times a week. Hence, they are concerned with the work schedule of the refuse collection problem, and not so much with the routing of the vehicles. They give some interesting discussions on how the problem changes when considering a system where the refuse is divided into types (paper, organic refuse etc.) that need to be handled separately.

2.4.4 Electric Meter Reading

Dror and Stern 1979 [59] consider the problem of routing electric meter readers. The problem occurs for municipal public service agencies which, in some countries, periodically have readers going from house to house to collect data for billing purposes. The readers are transported to the beginning of their route, works for a number of hours and are free to leave afterwards. To minimize labor cost, the companies want to minimize the dead-heading time, and thereby the number of workers. The problem can be modeled as a CARP-like problem since it requires service of streets, and the time each reader may work is limited. However, the solution to this problem are paths and not cycles, since they do not include connections to and from the Depot node. They solve the problem by constructing a giant tour which is in turn split, using simple forward splitting, into small tours corresponding to one reader.

2.4.5 Airline Scheduling

Many parts of Airline Scheduling can also be modeled as Capacitated Arc Routing. This includes the scheduling of planes to flight legs and the scheduling of crew. When doing this, each node in the graph corresponds to a destination and an arc in the graph corresponds to a flight leg. That way the problem is modeled as a directed arc routing problem. Many extra constraints must be taken into account when considering airline problems, including that the fleet of planes is not homogen and a complicated set of union rules for the crew. The costs in airline scheduling are high compared to the other applications that we have considered and so a reduction of even few percent in operation cost can improve revenue by millions of dollars.

Anbil et al. 1992 [9], Chu et al. 1997 [40] and Barnhart et al. 1997 [18] consider the crew scheduling problem and describe ways to model and solve the problem using a subproblem method. Hu and Johnson 1999 [95] present a primal-dual subproblem simplex, which extensively speed up the solution of the crew planning problem.

Goldsman et al. 2000 [82] present a stochastic model of the airline operation. They consider different ways of disrupting the operation and suggest various methods for recovering the schedules. They describe a simulation of airline operation called SimAir to simulate the schedules and the influence of the different recovery procedures on the longer term.

2.5 Variations of the CARP

As we have just seen, several of the applications of the CARP involve variations of the classical CARP. Here we will mention some of these variations, which have been treated in the literature.

2.5.1 Multi Depot CARP

The first variation of the CARP we will consider is the Multi Depot CARP (MD-CARP). The MD-CARP is defined as the classical CARP with the exception that now there are several Depots in which the vehicles are located and from which the tours must begin and end. The most common variation of MD-CARP is where each vehicle must return to the same depot as it started from. One could also consider the case where each vehicle just has to return to some Depot by the end of the day independently of which Depot the vehicle started from. In MD-CARP vehicles located in different Depots may have different capacity.

The MD-CARP frequently occurs in practice, though the problem is often considered as a classical CARP. When dealing with several of the applications considered in the previous section such as Street Sweeping, Winter Gritting, and Refuse Collection, the problem is naturally a multi Depot one when considering a large area. For ease administration the problems are partitioned into several single Depot problems on smaller geographical areas such as neighborhoods, towns, or counties. For each such smaller area there is an associated Depot where vehicle tours servicing that area begin and end, and where the administration of the service of that area is handled.

Cattrysse et al. 2002 [33] and Cattrysse et al. 2003 [34] consider the problem of dividing the initial area into smaller areas, called districts. They argue that the design of these districts are on a long-term planning level, whereas the construction of vehicle tours are on a short-term planning level. Each district should be a connected geographical area with clear boundaries. Furthermore, streets that naturally belong together, such as the streets in a small neighborhood, should be in the same district. Finally, the districts should be of similar size and the Depot should be located as central in the district as possible.

The authors approach the problem by first partitioning the network into small units of edges that should be in the same district. They then heuristically merge units to form the districts. The partitioning of the network into units are done by making the network Eulerian and making a cycle decomposition of the resulting network. In this way each cycle will correspond to a face of the graph. Their first strategy for merging units into districts is to simply assign each unit to the nearest Depot. They also consider this strategy when the units are simple edges. Finally, they consider a strategy where the assignment of the units are based on minimizing the lower bound of the required number of vehicles needed to service the edges of the resulting districts.

Amberg et al. 2000 [7] also consider a Multi Depot variation of the CARP. In contrast to the previous approach they do not consider the definition of districts as long-term decisions but let the districts be formed as a result of the routing strategy. Specifically, the vehicle tours are constructed first, and then the districts are formed as the streets serviced by vehicles emerging from a specific Depot.

The solution strategy that is used is quite unique for arc routing as the authors, after constructing a giant tour, transform the problem into an Arc-Constrained Capacitated Minimum Spanning Tree Problem (CMST). This problem is then solved heuristically, and the solution is improved by Local Search. Finally a Multi Depot CARP solution is derived from the CMST solution, and the resulting tours are improved by a simple route optimization procedure.

Ghiani et al. 2001 [78] consider a variation of the CARP, which they refer to as the Capacitated Arc Routing Problem with Intermediate Facilities (CARP-IF). The problem, which can be viewed as a variation of the MD-CARP, is defined as the CARP with one Depot, but has a set of nodes known as intermediate facilities, IF. The vehicles start in the Depot and must return to the Depot when done, but they can recharge their capacity in any of the intermediate facilities. For practical purposes the IFs can be dump sites for refuse, storage halls for salt for winter gritting and the like.

The authors present two lower bounds and two heuristics for the CARP-IF. The first lower bound notes that the RPP is a special case of the CARP-IF, and therefore uses a relatively tight RPP lower bound based on branch and cut to bound the CARP-IF. The second lower bound is a relaxation lower bound of an ILP formulation based on dead-heading variables. The first heuristic they persent for the problem is based on constructing an RPP-tour and splitting the tour in appropriate portions while connecting to the intermediate facilities. The second heuristic is based on solving the classical CARP in a modified network, transforming the solution to a CARP-IF solution in the original network and making some adjustments to restore feasibility.

2.5.2 CARP with Time Windows

The CARP with Time Windows (CARP-TW) is defined as the classical CARP with the extention that each demand edge has a Time Window associated within which the service of that particular edge must begin. In addition to costs, travel times are associated with all edges, and the tours must be constructed to respect the additional restriction, that all edges are serviced within their Time Window.

CARP-TW has not been studied much in the literature, but do occurs in some of the applications of arc routing. Flight legs in Airline Scheduling have a fixed departure time and can therefore be considered as having a time window of zero length. Street Sweeping, as presented by Bodin and Kursh 1978 [30], have time windows on the streets to be serviced. Finally, Eglese 1994 [63] consides the routing of winter gritters where some streets must be swept after two hours, other after four etc. This problem can clearly be considered as CARP-TW, where the time windows are rather wide.

In his PhD. Dissertation, Mullaseril 1997 [117] consideres a directed version of the CARP-TW. He consideres the problem from a heuristic point of view and also gives a transformation of the directed CARP-TW to the node routing problem, VRP-TW.

2.5.3 Other Variations of the CARP

The **Periodic CARP (PCARP)** is defined as the CARP, where a long time horizon is considered such that each demand edge requires service more than once. This problem occurs some places in refuse collection where each household is serviced two or three times a week on a rolling schedule. Here one must take into account that the problem may require a minimum and maximum number of days between each service of the same street. Lacomme et al. 2002 [105] introduce this problem formally and present a generic algorithm for PCARP, which is an extention of the algorithm by Lacomme et al. 2001 [104]. Chu et al. [39] present two lower bounds for PCARP which are both based on CARP lower bounds on a transformed graph.

The **Stochastic CARP (SCARP)** is suggested by Fleury et al. [72] to be similar to the classical CARP except from the demand on the edges being a random variable. This problem occurs in practice in problems such as refuse collection where the exact demand is not known. Fleury et al. [72] consider the quality of the solutions obtained for SCARP when algorithms for the deterministic CARP (the classical problem) are used. They explore how the robustness of the solution changes when the deterministic problem is solved with a slightly smaller fictive vehicle capacity. Fleury et al. 2004 [71] present a Memetic algorithm for SCARP which is an extention of the algorithm by Lacomme et al. 2004 [107] and compares the result to algorithms for the classical CARP based on the average demand.

Lacomme et al. 2003 [106] consider an extention of the CARP which they denote **E-CARP**. E-CARP is defined as the classical CARP with the extention to handle problems defined on mixed graphs (i.e. graphs containing both undirected edges and directed arcs), prohibited turns, and different costs depending on whether or not a link is serviced or simply traversed. The authors give a mathematical model for E-CARP which can handle these extra constraints.

Ulusoy 1985 [131] considers a version of the CARP where a vehicle includes a **fixed cost** if it is used and where the vehicles differ in capacity. The objective function therefore is to minimize the total travel cost plus the total fixed cost incurred by the use of vehicles. The author considers both a case with an unlimited number of each vehicle type and the case where the number of each vehicle type is bounded. An heuristic algorithm is presented that first constructs a giant tour and then splits the tour by solving a Shortest Path Problem taking into account the vehicle capacity and costs.

Lacomme et al. 2003 [108] consider a **Multi Objective CARP**. This problem is defined as the classical CARP where the objective is not only to minimize the total routing cost, but also to minimize the make-span, i.e. the length of the longest tour. With this objective the problem can be viewed as a mix between the CARP and the Min-Max K-Chinese Postman Problem, which is presented in Frederickson et al. 1978 [75]. The authors present a generic algorithm for solving the Multi Objective CARP.