Research Paper Business Analytics

Applications for the Vehicle Routing Problem

Jelmer Blok

Applications for the Vehicle Routing Problem

Jelmer Blok

Research Paper

Vrije Universiteit Amsterdam
Faculteit der Exacte Wetenschappen
Studierichting Business Analytics
De Boelelaan 1081a
1081 HV Amsterdam
January 4, 2016

Preface

This research paper is reviewing all the different variants of the Vehicle Routing Problem. It is written in the context of my study Business Analytics from the Vrije Universiteit, Amsterdam. From the period October 2015 until December 2015, I was working on this paper.

Hereby I want to thank my supervisor, Ger Koole (VU University) for his time and advice. Special thanks to Prof. Dr. Gromicho (VU University, ORTEC) who helped me a lot with explaining the Dynamic Vehicle Routing Problem with Time Windows.

Abstract

This paper reviews optimization methods and software products for efficiently solving Vehicle Routing Problem variants. First a brief introduction of all the VRP variants will be given.
Next different optimization methods will be explained. These methods have an important role in this paper.
Then the Traveling Salesman Problem (TSP) will be discussed. Two software products, the Concorde solver and the TSP solver in R, are being reviewed. The Concorde solver uses a Branch and Bound algorithm. The TSP solver uses an arbitrary insertion algorithm with two opt refinement. The Concorde solver is on average 3% better than the TSP solver in R.
After this the Vehicle Routing Problem (VRP) will be handled, where two software products will be reviewed and compared with each other. The software products are open source and each have a different solution method. The first software product is a VBA algorithm that uses an insertion heuristic with the improvement heuristics 2-opt, 3-opt and swap operations. The second, a genetic algorithm in Mathlab, uses a 2-opt improvement heuristic. The VBA algorithm has on average 7% better results.
The last part of this paper will discuss the Dynamic Vehicle Routing Problem with Time Windows. This problem is based on the VRP, but now all the customers have a certain time interval in which they would like to be served. There are two types of customers, known customers and new customers. The known customers are already scheduled into routes. New customers can call in, and the company must immediately specify a time window in which the start of the service will be provided.
The BARTOC method is a successful method for solving this problem, but it is outdated for today's optimization problems. The BARTOC method has only a time and distance constraint but today's problems have more constrains. Therefore it has to be extended so it can handle more constrains.

Contents

1 Introduction

The Vehicle Routing Problem (VRP) is an optimization problem that has been studied with much interest in the last five to six decades by mathematicians and computer scientists. The context of the VRP is to give an optimal set of routes for a fleet of vehicles to travel, in order to give service to a given set of customers.
The VRP was first discussed by the mathematicians George Dantzig and John Ramser in their famous paper The Truck Dispatching Problem [1]. Dantzig and Ramser made the first algorithmic approach to the VRP and applied it to petrol deliveries. In 1964 Clarke and Wright improved the approach of Dantzig and Ramser by using a greedy algorithm [2].
Today there are many variants of the VRP. The VRP can be extended with pickup and delivery (VRPPD), multiple depots (MDVRP), capacity limits (CVRP) and time windows (VRPTW). The VRPTW can be extended where new customers call in, and must be planned into the existing routes. The new customer receives an answer within five seconds. This problem is called the Dynamic VRPTW (DVRPTW). The applications for this problem are found in the areas of package delivery, traveling repairman's and distribution of products.
Since the growth of online shopping and environmental awareness, the demand for route optimization software increased. Today there are many software products available for all kinds of variants of the VRP, but which software products give the best solution and which method is used for solving this problem? The DVRPTW is one of the most difficult vehicle routing problems, since the complexity of the problem and the computation time has to be short. The Booking Algorithm for Routing and Timing Of Customers (BARTOC) is a method for solving the DVRPTW. It can provide an answer within the required five seconds. But what are the arguments that this method is not applicable for today's optimization problems and how can this method be adjusted so that it is applicable?
This paper reviews the different variants of the VRP, software products and methods for solving the VRP variants. Extra attention is given to the DVRPTW and their solution method.

2 Vehicle Routing Problem variants

As already mentioned in the introduction there are many variants of the VRP. Every company has different criteria specific to their optimization problem. They could have one vehicle or multiple vehicles, visiting one customer or multiple customer per route or have capacity limits. The VRP is a very wide problem.

2.1 Traveling Salesman Problem

The Traveling Salesman Problem (TSP) was first formulated in 1930 and is one of the most intensively studied problems in optimization. The context of this problem is to give an optimal route for one vehicle in order to travel to each node exactly once. This problem is computationally difficult, but a large number of optimization methods are known for solving this problem. These methods will be further discussed in section 3. Figure 1 and 2 give a graphical representation of the TSP problem.

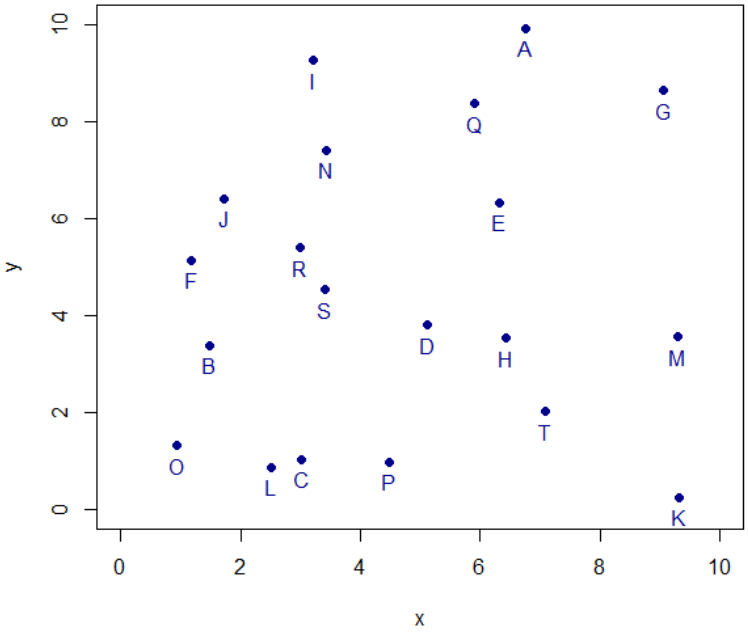


Figure 1: 15 randomly generated nodes

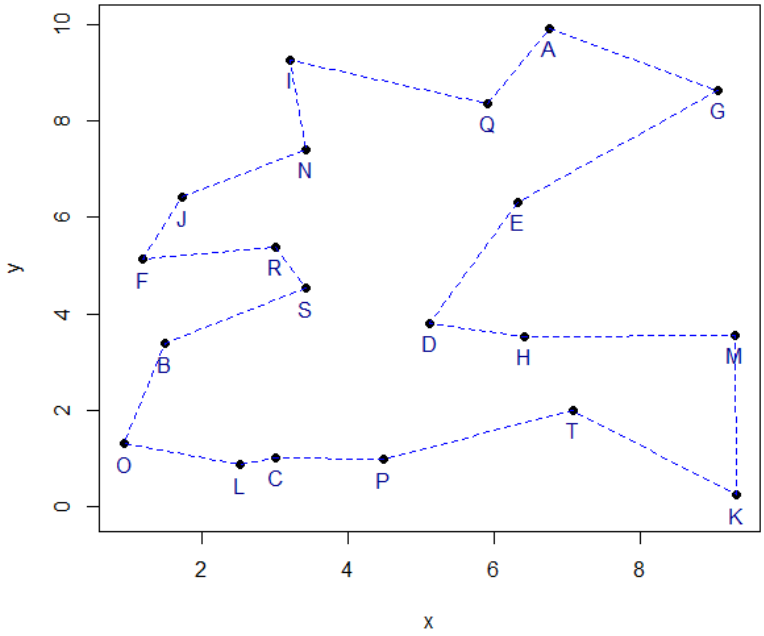
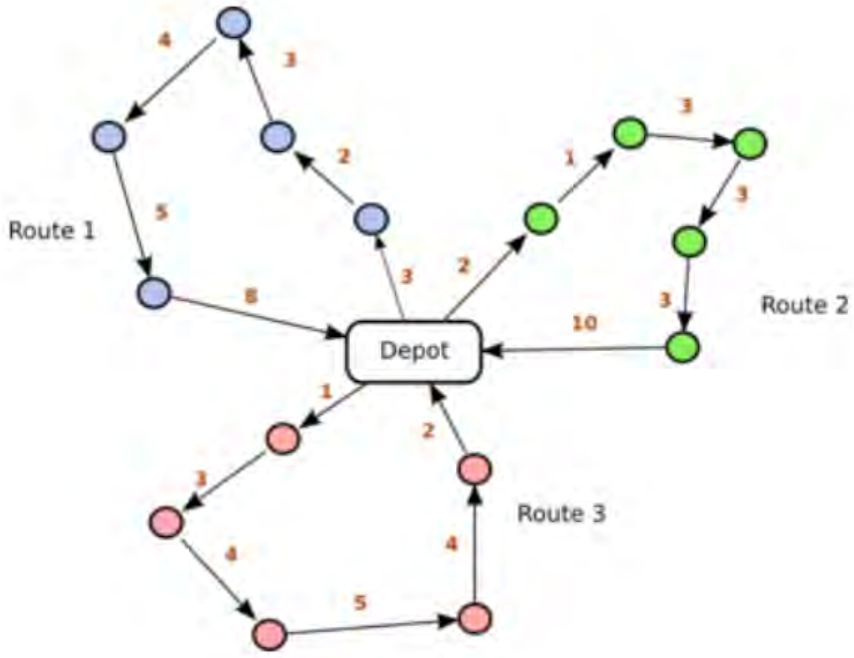


Figure 2: Optimal Tour

2.2 Vehicle Routing Problem

The Vehicle Routing Problem (VRP) was introduced by Dantzig and Ramser in 1959 [1]. This was the first algorithmic approach to this problem and was applied to petrol deliveries. In 1964, Clarke and Wright improved the approach of Dantzig and Ramser by using an effective greedy approach, called the savings algorithm. The VRP is a combinatorial optimization problem where multiple vehicles have to travel in order to visit each node exactly once. There are multiple objectives for the VRP i.e. minimizing costs, minimizing distance travelled or maximizing revenue. Therefore this is an important problem for companies in the fields of transportation, distribution and logistics. Figure 3 gives a graphical representation of this problem.

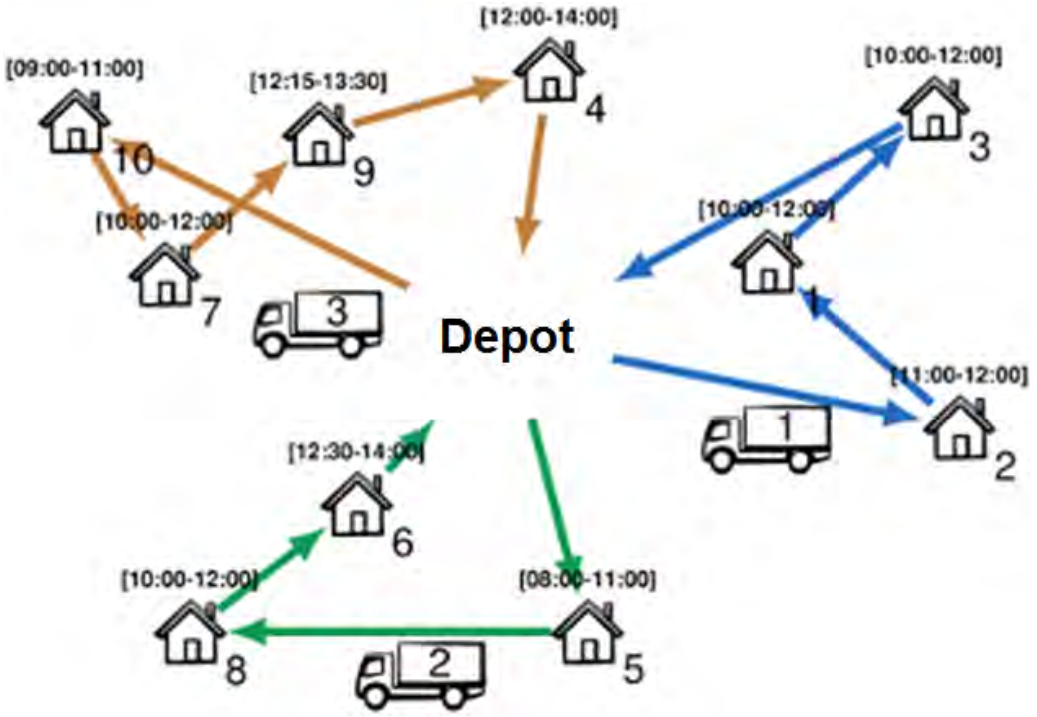
Figure 3: Vehicle Routing Problem



2.3 VRPTW

The VRPTW is a problem occurring often in service providing companies. Its an extension of the VRP problem, but for this problem the nodes (customers) have a certain time interval in which they would like to be served. The serviceman may arrive before the time window, but the node cannot be served before the time window opens. The serviceman has to wait until he can start providing the service. Of course it is also not allowed to arrive after the time window.
There are multiple objectives for this problem. One of the objectives is to design a set of routes that minimize the costs for each vehicle, such that each node is visited once in the preferred time window. Other objectives could be, minimize the number of vehicles, minimize travel distance or maximize volume delivered per kilometer. Figure 4 gives a representation of this problem.

Figure 4: Vehicle Routing Problem with Time Windows



2.4 DVRPTW

The Dynamic VRPTW is an extension of the VRPTW. But in this problem there is a distinction between two types of customers. There are ”known customers” and ”new customers”. The known customers are already planned into routes and the new customers call in. They have to be inserted into the existing routes. Within a short time period of five seconds, the system has to provide an answer in which the customer can be added into the existing route. For this problem the demand is known according to the Poisson Process.

3 Optimization methods

This chapter will describe what the different types of optimization methods are. The goal of an optimization method is to find an optimal or near optimal solution with a low computational effort. The effort can be measured by the time or memory it consumes by the method. There are a lot of optimization methods that are helpful for solving the VRP. One can uses them separately, or in combination with each other to try to get an even better solution. Several types of optimization methods will be described.

3.1 Exact method

Exact methods guarantee in finding an optimal solution. Four exact methods are described in this paper.

3.1.1 Brute Force

Brute Force is a method that computes all the possible solutions of an optimization problem. This method works well for a small number of nodes, but turns extremely slow when the number of nodes is large. Considering the TSP where all nodes are in connection with each other. If the total number of nodes equals N, then all the possible solutions are $N!$ When $N = 25$ there are 1.55^{25} possible solutions. This is nearly impossible for a computer to calculate and to find the optimum in an applicable time. The computation time is expected to grow exponentially with the instance size. The time for calculating the optimal solution of 24.978 cities is 85 CPU years [11].

3.1.2 Branch and Bound

Branch and Bound is one of the most successful exact approaches for the VRP. Branch and Bound uses a search tree with lower and upper bounds. It calculates candidate solutions and forms a rooted tree. The algorithm explores branches of this tree, which represent subsets of the solution set. Before enumerating the candidate solutions of a branch, the branch is checked against upper and lower estimated bounds of the optimal solution, and is discarded if it cannot produce a better solution than the best one found so far by the algorithm. The result is a limited depth of a search tree. The result in tour length compared with the Brute Force method, will be the same. Only difference between the Branch and Bound

method and the Brute Force method is that the Branch and Bound method will enumerate branch of the search tree, which result in a faster computation time.

3.1.3 Integer linear programming

Integer linear programming is an optimization technique. With this method the TSP can then be formulated as an integer linear programming problem. This method will result in a minimum tour length for the given nodes. Given is the distance between each node, c_{ij}. Let u_i be a sequence number in which city i visited. Let

$$x_{ij} = \begin{cases} 1 & \text{if there is a path from node } i \text{ to node } j \\ 0 & \text{otherwise} \end{cases}$$

The objective function is:

$$min \sum_{i \in N} \sum_{j \in N} x_{ij} c_{ij} \tag{1}$$

Subject to:

$$\sum_{i \in N} x_{ij} = 1, \forall j \in N \tag{2}$$

$$\sum_{j \in N} x_{ij} = 1, \forall i \in N \tag{3}$$

$$u_i - u_j + N x_{ij} \leq N - 1, \forall (i,j) \in N, i \neq 1, j \neq 1 \tag{4}$$

$$2 \leq u_i \leq N, \forall i \in \{2...n\} \tag{5}$$

$$x_{ij} \in 0, 1, \forall i, j \in N \tag{6}$$

Constraint 2 requires that there is an arrival from exactly one other node. Constraint 3 requires that there is a departure to exactly one other node. Constraint 4 and 5 enforce that there is only a single tour, and not two or more disjointed tours. Consider a tour that goes from node 3 to 4, $u_3 - u_4 + N x_{3,4} \leq N - 1$ $\rightarrow u_4 \geq u_3 + 1$. Similarly u has to increase by 1 along each edge of the cycle that does not include the begin vertex 1. (Assume that the tour begin in node 1). Therefore for a tour of length L that does not go through vertex 1, $u_3 \geq u_3 + L$ is a contradiction. Every tour must go through vertex 1. Together with the other constraints, it forces to create one tour [16], [17]. See figure 5 and 6 where constraints 4 and 5 are being explained in an example.

Figure 5: Example for the integer linear programming method

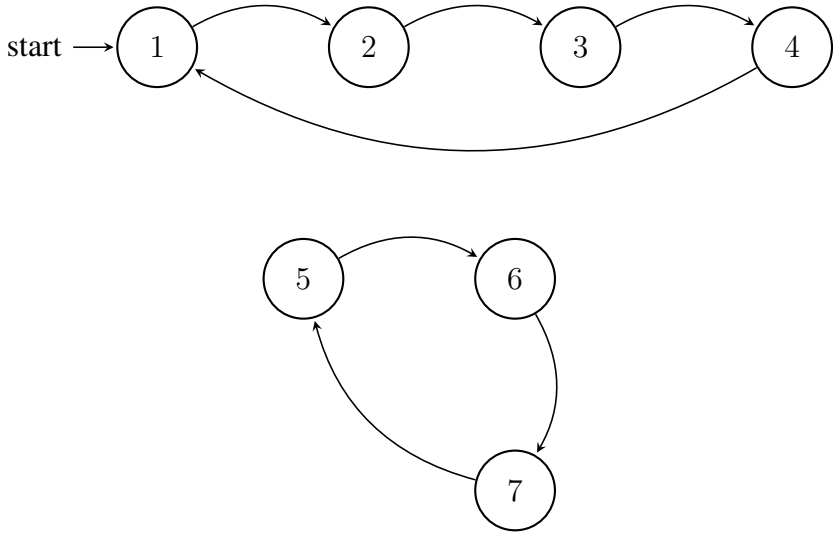
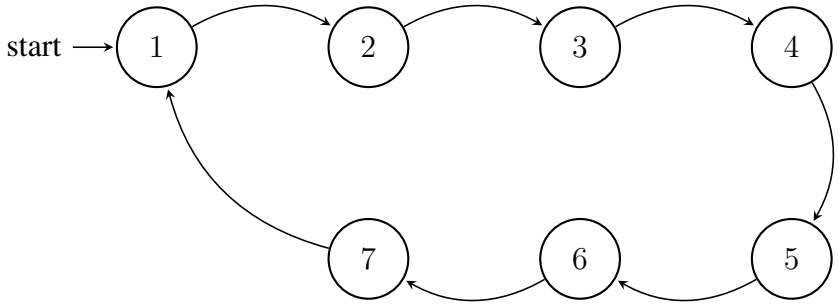


Figure 5 shows seven nodes. Nodes 1,2,3,4 and 5,6,7 are forming a subtour. Using constraints 4 and 5 will show that the result for this optimization problem is not feasible, since all the seven nodes should be into one single tour.
From node 2 to 3: $2 - 3 + 7 \leq 6$ ✓
From node 3 to 4: $3 - 4 + 7 \leq 6$ ✓
From node 4 to 5: $4 - 5 + 0 \leq 6$ ✓
From node 5 to 6: $5 - 6 + 7 \leq 6$ ✓
From node 6 to 7: $6 - 7 + 7 \leq 6$ ✓
From node 7 to 5: $7 - 5 + 7 \leq 6$ ✗

The result from node 7 to 5, does not satisfy constraint 4.

Figure 6: Example 2 for the integer linear programming method



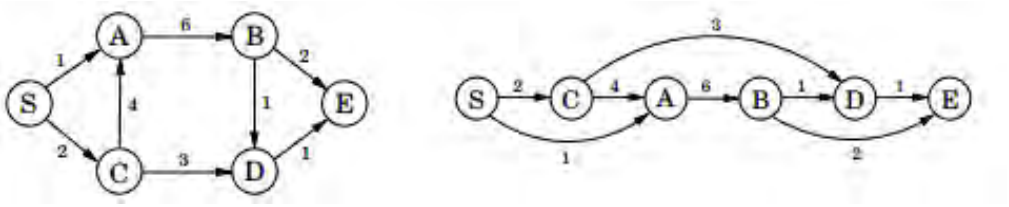
Using constraints 4 and 5 will show that the optimization problem presented in figure 6 is feasible, since all the seven nodes are into one single tour.
From node 2 to 3: $2 - 3 + 7 \leq 6$ ✓
From node 3 to 4: $3 - 4 + 7 \leq 6$ ✓
From node 4 to 5: $4 - 5 + 7 \leq 6$ ✓
From node 5 to 6: $5 - 6 + 7 \leq 6$ ✓
From node 6 to 7: $6 - 7 + 7 \leq 6$ ✓

The outcome of the objective function is the minimum tour length. This algorithm will present the same result as the Brute Force method, only the computation time is less.

3.1.4 Dynamic programming

Dynamic Programming is a method for solving a complex problem by breaking it down into a collection of simpler subproblems, solving each of those subproblems just once, and storing their solutions. The next time the same subproblem occurs, instead of recomputing its solution, one simply looks up the previously computed solution, thereby saving computation time and storage space. First the nodes must be arranged in a line so that all edges go from left to right, see figure 7. A DAG stands for Directed Acyclic Graphs.

Figure 7: A dag and its linearization [4]



If the distance from S to node D needs to be calculated then equation 7 has to be solved.

$$dist(D) = min(dist(B) + 1, dist(C) + 3) \tag{7}$$

A similar relation can be written for every node. If we compute these $dist$ values in the left-to-right order, a new $dist$ value of node x is then easily computed. In the end this method result in the shortest path from a given starting node [3]. The result of this method compared with the Brute Force method will be the same. The computation time will be less with the Dynamic Programming method since not all the possible routes needs to be calculated and the Dynamic Programming method stores the already found solutions of the subproblems.

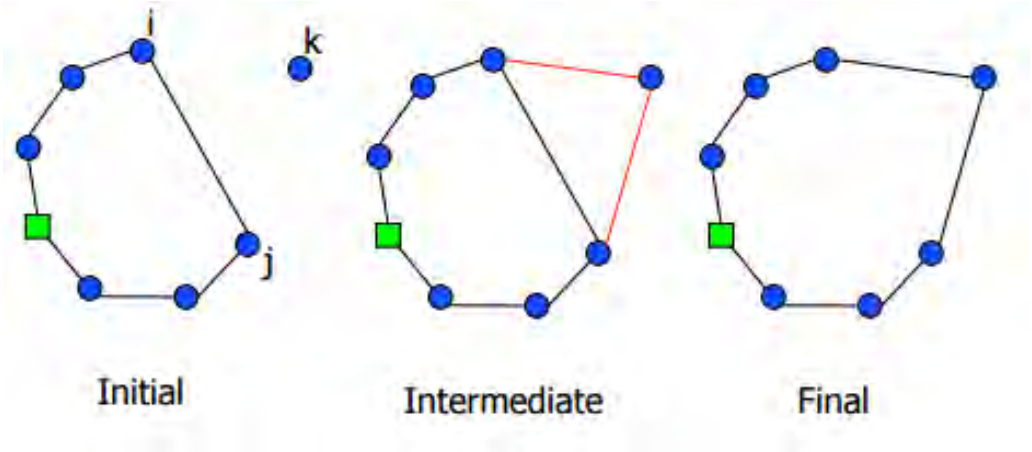
3.2 Heuristics

Many decision problems in business and economics can be formulated as an optimization problem. Most of these problems are too difficult to be solved exactly within a reasonable amount of time. Then heuristics becomes the method of choice. Two types of heuristics will be discussed, Construction heuristics and Improvement heuristics. The main difference between these two heuristics is that construction heuristics determines a tour according to some construction rules, but does not try to improve this tour. Improvement heuristic do try to improve a tour.

3.2.1 Construction heuristics

Two types of construction heuristics will be explained, saving heuristics and insertion heuristics. The insertion heuristic is used to check if an extra node can be added to a route [2]. See figure 8 for a representation of this method.

Figure 8: Insertion heuristic

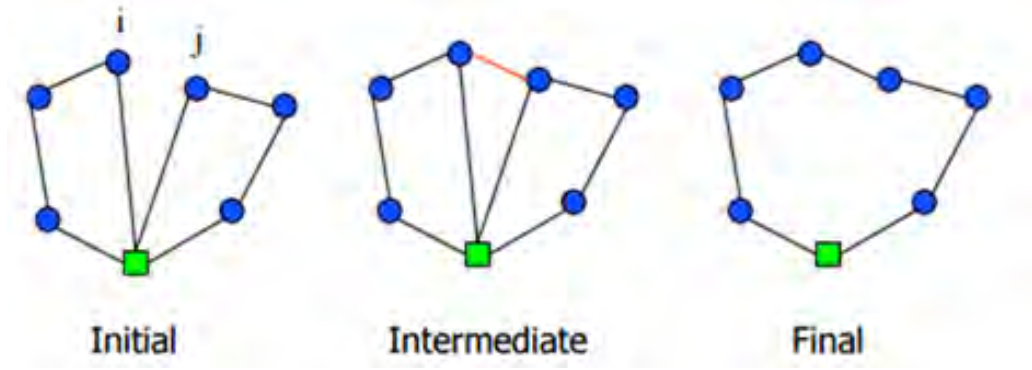


Formula 8 is used to calculate, between which two nodes the new city k can be inserted by checking all combinations of i, j. The minimum route length (I) is eventually chosen and city k can be inserted between i and j.

$$I_{ij} = C_{ik} + C_{jk} - C_{ij} \tag{8}$$

The Saving heuristic is used to calculate whether two routes can be added together. This method starts with an initial allocation of one vehicle to each customer. Then it calculates the possible saving if two routes are added together, with $I_{ij} = C_{0i} + C_{0j} - C_{ij}$ were 0 represents the depot. If I_{ij} has been calculated and the minimum has been chosen and i and j are not in the same tour, then i and j should be added together. See figure 9 for a representation of this method.

Figure 9: Saving heuristic

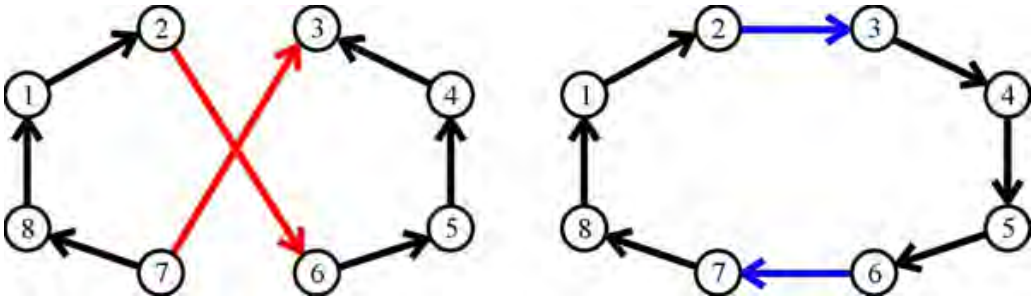


Other construction heuristics are Nearest Neighbor, Cheapest Insertion and Farthest Insertion.

3.2.2 Improvement heuristics

A well-known improvement heuristic is k-opt. This is a local search algorithm and its applies local changes in existing routes, to find the best solution. 2-opt and 3-opt are methods that will be briefly described and are very successful in improving the route.
The main idea about 2-opt is it will compare every possible combination of swapping two nodes, to check if there is a better solution possible. See figure 10 for a representation of this problem.

Figure 10: Example of the 2-opt heuristic [8]



3-opt heuristics involves deleting three arcs in a tour, then reconnecting the network in all other possible ways, and then evaluating each solution to find the optimum.

4 Traveling Salesman Problem

As mentioned in section 2, the Traveling Salesman Problem (TSP) is an NP-hard problem and covers the fields of optimization, operations research and computer science. The goal of the TSP is to give an optimal tour for one vehicle for a given set of nodes, where each node is only visited once. The objective of this problem is to find a route that minimizes the total distance travelled. A large number of exact methods and heuristics are known to solve this problem. The TSP has several applications other than only planning and routing. It appears in other sub-problems as well, such as soldering points, DNA fragments and astronomy.
The TSP can be modeled as an undirected weighted graph such that cities are the graph's vertices, paths are the graph's edges, and a path's distance is the edge's length. The minimization problem starts and finishes at a specific vertex.

4.1 Software

There are many software products available that can solve the TSP problem. In this paper two software products for the TSP will be tested. The Concorde solver and the TSP package in R. Both will be tested on how good their optimal solution is.

4.1.1 Concorde Solver

The Concorde solver is one of the best known and exact solvers today [7]. It uses a Branch and Bound algorithm to find the optimal solution. The Concorde solver has been extremely successful for large data sets. The largest data set contained 85,900 nodes. The required input is a list with x and y coordinates.

4.1.2 TSP package

The TSP package in R is written by Michael Hahsler [5]. It can create a route for a number of nodes with different types of methods. The required input is a list with x and y coordinates. The "optimal method" uses an arbitrary insertion algorithm with two opt refinement.

4.2 Results

In table 1 the results are shown of the several tests between the Concorde solver and the TSP solver. A data set is created where the nodes have a random x and y coordinate $\in [0, 10]$. Each software product calculates their optimal tour for the given data set. C represents the Concorde solver and R the R solver. L represents the tour length and T the computation time in seconds.

Table 1: TSP results table

# nodes	L(C)	T(C)	L(R)	T(R)
25	431	0,22	435	0,06
50	541	0,27	544	0,08
75	664	0,67	665	0,15
100	805	2,19	817	0,90
125	891	2,61	907	1,05
150	966	2,40	992	1,31
175	1029	1,87	1065	1,32
200	1096	2,89	1136	1,84
225	1133	5,36	1180	2,11
250	1185	8,63	1238	2,88
275	1230	13,47	1303	4,53
300	1295	14,84	1366	6,11

The Concorde solver gives a better solution for each set of nodes. When the number of nodes increases, the difference between both products increases as well. On average the Concorde solver gives a 3% better solution than the R solver. The computation time of the R solver is shorter than the Concorde solver, but its solution is worse.

5 Vehicle Routing Problem

The Vehicle Routing Problem (VRP) was introduced by Dantzig and Ramser in 1959 [1]. The VRP is a combinatorial optimization problem were a number of vehicles have to travel in order to visit each node exactly once. Where the TSP is applicable for one vehicle, the VRP have multiple vehicles.
Multiple objectives can be applicable for the VRP i.e., minimizing costs, minimizing distance travelled or maximizing revenue. The most common objective is minimizing costs, since this is the most relevant reason for companies to use the VRP software. But if an employee is paid per package he delivers, he wants to maximize the number of packages delivered. So multiple objectives are applicable, but it can differ for every situation. Therefore this is an important and often returning problem for companies in the fields of transportation, distribution and logistics.

5.1 Vehicle Routing Software

There are many software products available for solving the VRP. There is free open source software and commercial software. For this paper open source software will be discussed.
In mathlab Joshep Kirk developed a program for solving the VRP. This program uses a Genetic Algorithm to solve the VRP. Required input is an x and y coordinate and the number of servicemen available. The algorithm starts with a random solution for the given number of servicemen. The program starts with initial routes for each serviceman and uses 2-opt and swap operations to optimize the routes [9].
In Excel, Güneş Erdoğan developed an algorithm that solves the VRP. The most relevant input fields of this program are the address of the customers, the number of trucks and the average speed of the vehicles. The coordinates are found by the Bing search engine from Microsoft. In the next step the Euclidean distance matrix is calculated. After this step the algorithm will start. It uses a saving heuristic with 2-opt, 3-opt and swap operations. Swap operations is where two nodes on different routes are removed from their routes and inserted into the other route. The solution is shown for each vehicle separately to which customer and at what time it should arrive and depart. [10]

5.2 Results

For testing both software products a data set is created were the nodes have a random x and y coordinate $\in [0, 10]$, with the central depot located at the median [5, 5]. Both software products will be tested, with the same data set, for their optimal solution. The objective for this test is minimizing the total tour length. The results are presented in table 2, where E represent the VBA algorithm and M the Mathlab algorithm. T represents the time in seconds.

Table 2: VRP results table

# of Nodes	# of Vehicles	Route (E)	T(E)	Route (M)	T(E)
10	2	29.930	0.090	29.930	3
25	3	48.170	0.250	48.170	27
50	3	61.960	3.350	65.140	115
66	4	75.420	2.110	81.970	200
75	3	75.700	9.860	77.470	250
80	4	81.880	3.740	83.720	280
100	4	85.050	10.830	94.170	340
150	15	179.140	12.630	233.12	550

The algorithm programmed in VBA is more successful than the Mathlab algorithm. Overall the VBA algorithm is 7% better. The VBA algorithm uses more optimization heuristics. Especially the 3-opt heuristic is useful to use for a large number of nodes in a route. This indicates that the more heuristics are used, the better the result will be.

6 The Dynamic VRP with Time Windows

The VRPTW is an extension of the VRP problem, but for this problem the nodes (customers) have a certain time interval in which they would like to be served. The VRPTW is NP-hard, so it requires a heuristic solution approach. All the parameters, such as demand locations and time windows, are assumed to be known. Each customer i has a time window $[a_i, b_i]$. The serviceman must arrive before b_i. He can arrive before a_i, but then the customers cannot be served. The serviceman has to wait until he can start with providing the service. There are two types of time windows, soft time windows and hard time windows. Soft time windows allow delays, but these delays will be punished with a penalty. Hard time windows do not allow any delays.
There are multiple objectives for this problem. One of the objectives is to design a set of routes that minimize the costs for each vehicle, such that each node is visited once in the preferred time window. Other objectives could be, minimize the number of vehicles, minimize travel distance or maximize volume delivered per mile. Because of the difficulty of the VRPTW and its wide applicability to real-life situations for service providing companies, many heuristic solution techniques capable of producing high-quality solutions in limited time have been proposed, see [12], [13].

The Dynamic VRPTW (DVRPTW) is based on the VRPTW but for this problem there are two types of customers, "known customers" and "new customers". The known customers are already scheduled into a route, but new customers are not. When a new customer call comes in, the company must immediately specify a time window in which the start of the service will be provided. Therefore the software must have a fast response time. There are two types of methods that will be discussed for solving this problem.

6.1 BARTOC method

The Booking Algorithm for Routing and Timing Of Customers (BARTOC) is a heuristic approach for dispatching servicemen and is dated from 1995 [14]. The time windows for this method consist of a length of two hours in which the customer can be placed. The customers are interested in a narrow time window, while the company is interested in a wide time window for flexibility.
The goal of the BARTOC method is to define for a fixed service level a near optimal strategy of route design and time window settings so that the total distance

travelled is minimized over the given time horizon. The customers are given an immediate answer which requires that the system needs to be interactive, fast response time and preferably usable on a personal computer.

6.1.1 Heuristic

The BARTOC method is based on a heuristic called, the cluster-first route-second principle and has four phases. After these phases a new customer is inserted into the existing routes, or is transferred to the next planning period if it could not fit into the existing routes.

Phase 1: The nodes are divided into N districts which is equivalent to the number of vehicles of the planning horizon. Each district has his own ”route center” or ”seed point”. It has not been defined how these seed points are calculated, but I assume k-means clustering could be used for this.

Phase 2: For each new customer the seed points are recalculated on all the known customers locations. This means that all seed points locations are influenced by a new customer. After this N_2 clusters are defined in which the new customer can possibly be added.

Phase 3: The new customer is added to the route with the insertion heuristic of Clark and Wright. All the routes that are selected in phase 2, N_2, are checked, and the route with the shortest new route length is chosen. If a new customer cannot be added, it will move to the next planning period.

Phase 4: In this phase the time windows and the sequence is checked.

Madsen (1995) showed that the BARTOC method compared with a standard savings based sequential insertion heuristic (Mole and Jamison [15]) performed well. This insertion heuristic had other criteria than the BARTOC method namely; the customers were all known, no time windows were imposed and the customers were not grouped in days. The results of this comparison were of good quality according to several evaluation criteria. All customers could be planned into the routes and the system had a fast response time. The computation time was less than two seconds and 20 different routes were planned. The BARTOC method was tested on an IBM 3033 computer which is dated from 1978. Although only four tests were performed in which the BARTOC method showed on average 1.6%

worse results in tour length than the insertion heuristic, Madsen considers the BARTOC results as reasonable and of good quality. (Note that the goal of the BARTOC method is to give a near optimal route).

6.2 ORTEC DVRPTW

In the Netherlands, ORTEC is a leading company in commercial optimization software. They have software for solving all different types of VRP variants. One of their largest clients in PostNL, a leading Dutch company in packages delivering. ORTEC have several software programs to solve the DVRPTW. One of the methods they use is the insertion method with multiple improvement heuristics. Since the number of constrains for optimization problems is very high, it is impossible to give a solution within a short time period. This is also mentioned in a study by the university in Denmark [6].

"Even though the rapid development in computer hardware along with a long line of improvements in optimization software has pushed the boundaries of what is possible to solve in a reasonable amount of time, it is still computationally hard to solve real-life static VRP's. This is mainly due to the wide variety of side constraints that are often present in a real-life routing problem." (Larsen, 2000)

Therefore the software of ORTEC, first makes an estimation if the new customer can be added to a route. After this estimation the optimization will start. With this method ORTEC satisfy the desire of the customer to give a solution within a short time period of five seconds.

7 Conclusion

The goal of this paper was to find a good optimization method and software products for different variants of the VRP problem. The TSP, the VRP and the DVRPWT were discussed.
The TSP problem can be solved best with the Concorde solver. It was already known that this solver was one of the best, but this paper has shown that it performed better than the solver in R with a 3% better tour length.
The VRP problem has many applicable software products. Two open source products were tested and the VBA algorithm preforms better, since it had more improvement heuristics.
The BARTOC method worked very well in 1995, but today's optimization problems are much more complex with more constrains. Larsen (2000) showed that an exact approach of the DVRPWT is very complex and not realistic, and it cannot provide an answer within the required time. This vision was also endorsed by Prof. Dr. Gromicho (VU University and ORTEC). Therefore ORTEC check first with an estimation if the new customer can be added to a route. After this estimation the optimization starts.
The BARTOC method was tested on an IBM computer from 1978. Computers are much faster today, which means that the BARTOC can be extended and is still be able to compute a solution in the preferred time. The results of the BARTOC method were of good quality according to several evaluation criteria. All customers could be planned into the routes and the system had a fast response time. In general, if the BARTOC method can be extended so it can handle more constrains and more improvement heuristics, I believe that this is a good method for today's optimization problems. Correctly balance the number of heuristic used and the associated computation time, is something every developer must keep in mind. Since an algorithm that has a computation time of several days is not usable for today's optimization problems.

References

[1] Dantzig, George Bernard; Ramser, John Hubert (1959). "The Truck Dispatching Problem". Management Science 6 (1): 80–91.

[2] G. Clarke and J. W. Wright (1969), "Scheduling of vehicles from a central depot to a number of delivery points," Operations Research, vol.12 , pp. 568–581.

[3] S. Dasgupta et al (2006), "Algorithms", pp. 170-175

[4] S. Dasgupta et al (2006), "Algorithms", pp. 161

[5] Hahsler, Michael; Hornik, Kurt (2007), "TSP – Infrastructure for the Traveling Salesperson Problem", Journal of Statistical Software 23 (2): 1–21.

[6] Larsen, Allen (2000), "The Dynamic Vehicle Routing Problem", Ph.D. Thesis Technical University of Denmark

[7] Concorde Solver, Retrieved from `http://www.math.uwaterloo.ca/tsp/concorde/downloads/downloads.htm`, 11-2015

[8] `http://www.devx.com/dotnet/Article/33574/0/page/3`, 12-2015

[9] `http://www.mathworks.com/matlabcentral/fileexchange/13680-traveling-salesman-problem-genetic-algorithm`, 11-2015

[10] http://verolog.deis.unibo.it/vrp-spreadsheet-solver, 11-2015

[11] Carlos Cotta,Jano van Hemert, "Recent Advances in Evolutionary Computation for Combinatorial Optimization", 2008

[12] O. Braysy and M. Gendreau, "Vehicle routing problem with time windows, part I: Route construction and local search algorithms". Transportation Science, 39(1):104–118, 2005.

[13] O. Braysy and M. Gendreau, "Vehicle routing problem with time windows, part II: Metaheuristics". Transportation Science, 39(1):119–139, 2005.

[14] O. Madsen, "A heuristic method for dispatching repair men", Annals of Operations Research 61: 213-226, 1995

[15] R.H. Mole and S.R. Jamison, A sequential route-building algorithm employing a generalized savings criterion, Operational Research Quarterly 27(1976)503-511.

[16] https://support.sas.com/documentation/cdl/en/ormpug/63352/HTML/default/viewer.htm#ormpug_milpsolver_sect020.htm, 01-2016

[17] G. Pataki, Teaching Integer Programming Formulations Using the Traveling Salesman Problem, Slam review, Vol. 45,No. 1,pp. 119