ERASMUS UNIVERSITY ROTTERDAM

Bachelor thesis Econometrie & Operationele Research specialization Quantitative Logistics

An extended Max-Min Ant System evaluated on the Split Delivery Weighted Vehicle Routing Problem

Abstract

This thesis discusses a relatively new type of vehicle routing problems, the Split Delivery Weighted Vehicle Routing Problem (SDWVRP), where both split deliveries of goods are allowed and the vehicle weight is included in the objective function. The SDWVRP is a NP-hard problem, which is why it is solved by a Max-Min Ant System (MMAS) heuristic, a more recent development in ant colony optimization. The goal of this thesis is first of all to revise earlier obtained MMAS results for the SDWVRP and compare them with related vehicle routing problems. Then, a candidate list and a 2-opt local search heuristic are performed on the MMAS algorithm to investigate whether cost savings are possible. Computational results indicate that combining both a candidate list and a 2-opt algorithm accomplish a cost reduction of 9.8% on average for the SDWVRP. Furthermore, the differences in results between the MMAS for the SDWVRP and the WVRP, which is the same model without the possibility for split deliveries, are small.

Erik van der Heide (481579)

Erasmus School of Economics

Supervisor: Y.N. Hoogendoorn

Second assessor: dr. R. Spliet

Erasmus School of Economics

July 5, 2020 - Erasmus University Rotterdam

Contents

1 Introduction

Vehicle Routing Problems (VRPs) belong to a class of well-studied combinatorial optimization problems. VRPs generalize the Traveling Salesman Problem, which aims to find the shortest path between a fixed set of nodes. The first VRP, called the Truck Dispatching Problem, appeared in a study on petrol deliveries by Dantzig and Ramser (1959). In the standard VRP, a number of vehicles have to satisfy the demand of various customer nodes, where all vehicles start and end in a single depot and the total traveled distance is minimized. When every vehicle has a limited capacity of goods to deliver instead of an infinite capacity, it generalizes to the *Capacitated Vehicle Routing Problem* (CVRP). For the CVRP, every vehicle has the same capacity, which is at least as large as the highest customer demand. These kind op VRPs have several practical applications, such as package delivery, waste collection and health care logistics, to name a few (Golden et al. 2010).

Over the years, many generalizations to the traditional VRP have been made. One of them is the relaxation of the traditional assumption that every customer is served by exactly one vehicle, which allows for a customer to be served by multiple vehicles. This is called a split delivery, which means that multiple vehicles serve a part of the demand of a customer. Dror and Trudeau (1989) were the first ones to introduce split deliveries and came up with the *Split Delivery Vehicle Routing Problem* (SDVRP). Archetti et al. (2006) show that by allowing for split deliveries a considerable cost reduction of at most 50% could be accomplished. Nevertheless, according to Archetti and Speranza (2012), the SDVRP is a challenging problem which can only be solved exactly with problem instances which consists of less than 30 customer nodes. Another finding of Archetti et al. (2008) is that the distribution of the customer demands have a much bigger influence on the cost savings than the customer location distribution. When the mean of the customer demands is just over half the vehicle capacity and the standard deviation of these demands is low, the cost savings by introducing split deliveries are potentially the highest. These cost savings are mainly achieved because of the reduction of the total number of delivery routes.

A second generalization of the VRP is to take the vehicle weight into consideration. In the standard CVRP, only the total traveled distance by all vehicles is minimized and the vehicle weight is not considered in the cost minimization. However, it can be of interest to include total vehicle weight into the minimization as well. The total weight of a vehicle is a combination of the fixed weight of a vehicle itself and the loaded cargo weight, which varies along a vehicle's route. Tang et al. (2013) mention the *Weighted Vehicle Routing Problem* (WVRP), for which the objective is to minimize the product of the vehicle's traveled distances and the corresponding vehicle weights. A real-life example is shown by Zhang et al. (2012), where a toll-by-weight regulation for China Expressway imposes costs on the vehicle weight. Furthermore, there are various types of problems in the literature which take the vehicle weight into consideration. For instance, the Demand Weighted VRP (Camm et al. 2017) for which the vehicle weight equals the number of passenger in the vehicle, or the Pollution-Routing Problem (Bektaş and Laporte 2011) where the vehicle weight is taken into account to minimize, among others, greenhouse emissions.

Combining both the split deliveries and vehicle weight generalizations to the CVRP, Tang et al.

(2013) introduce the *Split Delivery Weighted Vehicle Routing Problem* (SDWVRP). Their research shows the effectiveness of including load weight in the SDVRP on the one hand, as well as valuable cost savings achieved by including split deliveries in the WVRP on the other hand.

Tang et al. (2013) propose an exact mathematical formulation for this problem, but use a heuristic to solve this NP-hard problem. The heuristic they use belongs to the ant colony optimization algorithms. These algorithms are part of a big class of genetic algorithms, which are often used to find solutions for search and optimization problems. The idea of ant colony optimization is based on the natural behavior of ants searching for efficient routes to find food (Bell and McMullen 2004; Bullnheimer et al. 1999). Ants communicate with each other using *pheromones*. The more ants walk on a certain path, the more pheromone is left on this path, which increases the probability for the next ant to travel this path. After a while, shorter routes to rich food sources get preferred over less desirable longer routes. However, because of the random probabilistic behavior of ants, alternative routes are still considered. In this way, the natural intelligence of ants is used to solve complex routing problems.

The first ant colony optimization algorithm was the Ant System by Dorigo et al. (1991), originally performed on the Traveling Salesman Problem. For small instances, the results were promising, but larger instances were solved much more efficiently by well-studied metaheuristics like Simulated Annealing and Tabu Search. A similar conclusion holds for the CVRP (Bell and McMullen 2004). However, improvements on ant colony optimization methods are made over the years. For instance, Stützle and Hoos (1997) proposed the Max-Min Ant System, which imposes that pheromone levels should be limited to a certain range. This algorithm is less likely to have an early convergence and investigates currently found optima more actively. A computational study by Stützle and Hoos (2000) was done where the Max-Min Ant System was performed on the Traveling Salesman Problem, which showed that the results greatly improved compared with the original Ant System.

Tang et al. (2013) introduce an adapted *Max-Min Ant System for Split Deliveries* (MMAS-SD). They extend the original MMAS framework with a decision rule for split deliveries. The focus of this thesis is to implement the MMAS-SD algorithm and verify if similar computational results as Tang et al. (2013) are obtained. Furthermore, simplified versions of the MMAS-SD are implemented and compared with the original MMAS-SD. Next to that, the influences of different values of the fixed vehicle weight are tested. In addition, the aim of this research is to investigate whether further improvements on the MMAS-SD heuristic can be made to achieve a greater cost reduction. This is done in two ways: with a candidate list and a 2-opt local search algorithm. This thesis will demonstrate that a combination of a candidate list and a 2-opt algorithm added to the MMAS-SD algorithm improves the solution quality the most.

The remainder of this thesis is organized as follows. In Section 2, the Split Delivery Weighted Vehicle Routing Problem is explained and illustrated. Thereafter, in Section 3 the Max-Min Ant System for Split Deliveries algorithm is described as well as related MMAS algorithms, the candidate list structure and the 2-opt local search algorithm. Then Section 4 presents computational results and finally Section 5 gives the conclusion and topics for further research.

2 Problem description

This section first of all provides a complete description of the Split Delivery Weighted Vehicle Routing. Then, an illustration of the SDWVRP is given.

2.1 Description Split Delivery Weighted Vehicle Routing Problem

Following Tang et al. (2013), the SDWVRP is defined on a graph $G = (V, E)$. Here, $V = \{v_0\} \cup V_c$, where v_0 represents the depot and $V_c = \{v_1, v_2, \ldots, v_n\}$ is the set of customer nodes. Correspondingly, $E = \{(v_i, v_j) \mid (i, j) = 0, 1, 2, \ldots, n, i \neq j\}$ is the set of arcs. The parameter d_{ij} describes the Euclidean distance between node v_i and v_j, $(i, j) \in E$. Each customer i has a corresponding demand $q_i \geq 0$. Every vehicle k has a fixed weight M and capacity Q. The total vehicle weight is between M and $(M + Q)$ in every part of the solution. A total of K vehicles are needed to construct a complete solution.

A vehicle starts a route at the depot and ends this route at the depot. Starting from the depot, the vehicle traverses a sequence of customer nodes and its cargo weight changes on the way. If a vehicle performs a split delivery, the vehicle serves only a fraction of a customer's demand. In this case, one or possibly multiple vehicles serve the remaining demand fraction. When all vehicles together have served the demand of all customers, a complete tour is created. In the remainder of this thesis, we will continue to refer to a *route* as a trip from and to the depot by a single vehicle and a *tour* as a complete solution to the SDWVRP.

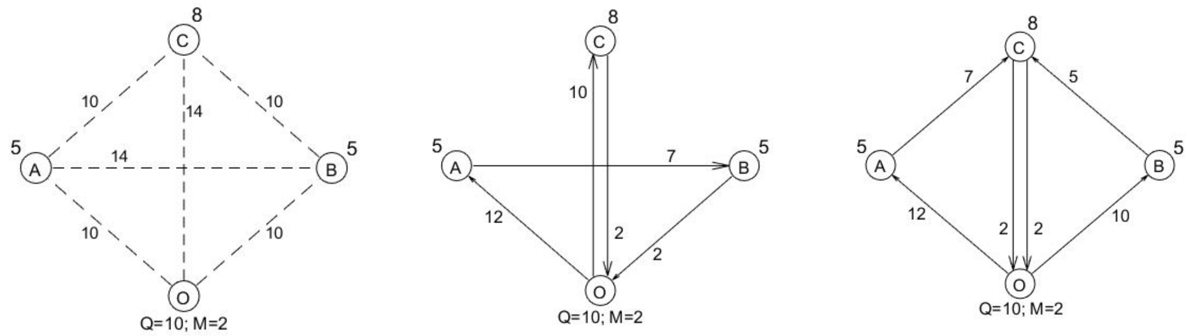
Several assumptions are made for the SDWVRP. First of all, distances d_{ij} are symmetric and comply with the triangle inequality. Next, there are infinitely many vehicles available to serve all customers. Also, no customer demand is greater than the vehicle capacity Q. Even though this is not a strict requirement to obtain a feasible solution in accordance with the SDWVRP, we always assume the vehicle capacity to be greater or equal than any individual demand. Further, the vehicle weight M is taken into account to make sure the weight cost for a vehicle returning to the depot is unequal to zero. Finally, every customer is completely served by at least one vehicle. and the total demand served by every vehicle does not exceed its capacity.

The objective of the SDWVRP is to minimize both the total travel cost in terms of distance as well as the vehicle and cargo weight during the transport. This is done by multiplying the distance d_{ij} with the total vehicle weight of vehicle k when this vehicle traverses arc $(i, j) \in E$. The sum over all these products of traversed arcs by all vehicles is minimized.

A complete mathematical formulation of the SDWVRP can be found in Tang et al. (2013). This model is highly non-linear and the problem is NP-hard, because it is a generalization of the SDVRP. Also, they provide some theoretical properties in which cases split deliveries are desirable. To solve the SDWVRP, Tang et al. (2013) use an ant colony optimization algorithm. They extend the MMAS framework with a decision rule for split deliveries based on their theoretical properties and come up with formulas for the minimum and maximum pheromone levels.

2.2 Illustration Split Delivery Weighted Vehicle Routing Problem

To illustrate split deliveries and its potential cost savings, an example is created on an small graph, shown in Figure 1a. This graph contains four nodes, where node O represents the depot. The vehicle capacity Q is set to 10 and the vehicle weight M to 2. The demands of the customers nodes are $q_A = q_B = 5$ and $q_C = 8$. The set of distances equals $d_{OA} = d_{AC} = d_{OB} = d_{BC} = 10$ and $d_{OC} = d_{AB} = 14$. The distances are symmetric, so for example it holds that $d_{OA} = d_{AO}$.



(a) Graph with distances (b) WVRP solution (c) SDWVRP solution

Figure 1: An example graph with a WVRP and SDWVRP solution

Figure 1b shows a solution to the WVRP problem, which means that split deliveries are not allowed. The two routes $\{O, A, B, O\}$ and $\{O, C, O\}$ are constructed, where in Figure 1b the total vehicle weight on the arcs are displayed next to the arcs. The weighted objective function value equals

$$C_{WVRP} = \big((10+2)\cdot d_{OA} + (5+2)\cdot d_{AB} + (0+2)\cdot d_{BO}\big) + \big((8+2)\cdot d_{OC} + (0+2)\cdot d_{CO}\big) = 406.$$

Figure 1c shows the best way to perform a split delivery in accordance with the SDWVRP. In this case, a split delivery is performed on node C, which means that this node is visited twice. In the first route 5/8 of node C's demand is satisfied and in the second route the remaining 3/8 demand is satisfied. In this case, the objective function value equals

$$\begin{aligned} C_{SDWVRP} = &\big((10+2)\cdot d_{OA} + (5+2)\cdot d_{AC} + (0+2)\cdot d_{CO}\big) \\ &+\big((8+2)\cdot d_{OB} + (3+2)\cdot d_{BC} + (0+2)\cdot d_{CO}\big) = 396, \end{aligned}$$

which is an improvement relative to the WVRP solution.

Two concluding remarks from the examples in Figure 1 have to be made. First of all, both Figure 1b and 1c do not show the optimal solution to their problems. In both cases, the optimal solution would be to perform three routes, where every route start in the depot, goes to one customer node and directly returns to the depot, which results in an objective function value of 348. Nevertheless, the example solutions in Figure 1b and 1c are chosen because they are realistic solutions to the MMAS algorithm as will be described in Section 3. Secondly, what the best solution is depends on the assumed vehicle weight M. If, for example, M is set to 5, the SDWVRP solution would be worse than the WVRP solution. However, the examples in Figure 1 are just for illustrative purposes.

3 Methodology

This section explains the Max-Min Ant System for Split Delivies as introduced by Tang et al. (2013), which seeks to find a feasible solution for the SDWVRP. In this algorithm, a single ant represents a complete tour, meaning that one ant creates a complete solution. Every ant incrementally constructs routes from and to the depot until all nodes are visited. In every iteration of the program, m such ants create a solution. The same notation is used as in Section 2.1. In addition, MMAS algorithms for related VRPs are discussed, as well as the candidate list and 2-opt local search extensions.

3.1 Max-Min Ant System for Split Deliveries

Let T_k be the 'tabu list' for ant $k \in \{1, \ldots, m\}$. This is the list of customers which are fully served by ant k at some point in a partial solution and do not have to be visited again. When node j is fully served, its demand q_j is restored to zero before it is added to T_k. Every ant chooses which paths between successive nodes to travel based on two criteria: the pheromone level and the visibility of the path. The pheromone level between node i and j is denoted by τ_{ij} and is updated in every iteration of the program, which is explained in Section 3.1.3. The visibility is denoted by η_{ij} and defined as $\eta_{ij} = q_j/d_{ij}$, which indicates that node j with a higher demand and closer to node i has a higher probability to be chosen. While constructing a solution, the ant will either fully serve a demand node if possible (see Section 3.1.1), or perform a split delivery otherwise (see Section 3.1.2).

3.1.1 Choosing a full demand node

When an ant starts the first route, its existing load is initialized as $Q_k^{left} = Q$. After serving node j completely, this is updated as $Q_k^{left} \leftarrow Q_k^{left} - q_j$. Now for the current ant k, let V_c^k be the set of customer nodes for which $q_i \leq Q_k^{left}$, $i \in V_c$. These are the customer nodes for which ant k is able to fully serve one of them at some point in a partial solution. This demand could be a fraction of the initial full demand of a node in case a split delivery was performed on this node earlier on in the algorithm. Set V_c^k is needed to choose full demand nodes.

At a particular stage of the tour, the next node is chosen either by picking the most favorable path or choosing a node based on a probability distribution function. This is called the *pseudo-random proportional rule* as proposed in the Ant Colony System by Dorigo and Gambardella (1997). A random variable $q \sim$ Uniform[0,1] is drawn and compared with a predefined parameter $q_0 \in (0, 1)$. A next node j is chosen if T_k does not contain all nodes, i.e. there is at least one feasible next node. If $q \leq q_0$, the ant currently at node i chooses the next node j according to (1).

$$j = \text{argmax}_{u \in V_c^k \setminus T_k} \{\tau_{iu}^{\alpha} \cdot \eta_{iu}^{\beta}\} \tag{1}$$

In (1), α and β are the influences of respectively the pheromone level and visibility on the optimal choice. If, on the other hand, $q > q_0$, the next node j is chosen based on formula (2).

$$\mathbb{P}[\text{node } j \text{ after } i] = p_{ij} = \frac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_{l \in V_c^k \setminus T_k} \tau_{il}^{\alpha} \cdot \eta_{il}^{\beta}} \quad \text{for } j \in V_c^k \setminus T_k \tag{2}$$

A random variable is drawn from the probability distribution (2), which will be the next visited node. The higher the pheromone and visibility of a feasible node, the higher the probability that this node will be chosen. When the next node j is chosen it gets added to the tabu list T_k. If the ant has no load left, it returns to the depot, otherwise it will choose a next node to visit.

3.1.2 Choosing a split delivery

In the MMAS-SD, the ant will return to the depot either when its full load is delivered or when all nodes are served. In the case that an ant still has some load left, but cannot serve a complete customer anymore, a different decision rule (3) is followed to perform a split delivery.

$$j = \text{argmin}_{s \in V_c \setminus T_k} \{(Q_k^{left} + M) \cdot d_{is} + M \cdot d_{s0}\} \tag{3}$$

Two criteria are minimized at the same time here: the remaining load and weight per distance to the next node and the weight per distance from that node to the depot thereafter. After node j is visited, the demand of node j gets updated: $q_j \leftarrow q_j - Q_k^{left}$. For example, when node j has a demand of 35 and ant k could only satisfy 20 units of this demand, the updated demand of node j equals 15. The node is not yet added to the tabu list T_k, because some part of the demand still has to be satisfied at another route. After node j is visited, Q_k^{left} will be zero. Ant k will return to the depot, where Q_k^{left} is restored to Q and a new route is set up with the remaining nodes available.

3.1.3 Updating pheromone levels

In every iteration of the program a total of m tours are created. After every m created tours, the pheromone levels are updated only by the ant which generated the least cost tour. This value is limited to $[\tau_{min}, \tau_{max}]$ according to (4), (5), as was proposed by Stützle and Hoos (2000).

$$\tau_{max} = 1/C^{best} \tag{4}$$

$$\tau_{min} = \frac{\tau_{max}(1 - \sqrt[n]{0.05})}{(n-2)\sqrt[n]{0.05}/2} \tag{5}$$

Here, C^{best} stands for the cost of the best found solution. When all ants have finished their tours, the next iteration of the program starts. The pheromone levels get updated following (6).

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij}^{best}(t) \tag{6}$$

In (6), t equals the iteration number of the program. To ensure that the effect of pheronomone does not get too strong, the parameter $\rho \in (0, 1]$ is used. According to Tang et al. (2013), this is called the *rate of pheromone evaporation*, which has to avoid an early convergence to a suboptimal local solution. $\Delta\tau_{ij}^{best}(t)$ is defined as $1/C^{best}$ if arc (i, j) belongs to the best tour and 0 otherwise. The best found solution can either be chosen as the *iteration-best* solution, i.e. the best found solution after each full iteration of the program, or as the global best solution, which is the best solution found so far. In addition, Stützle and Hoos (2000) suggest to reinitialize the pheromone trails if no improved solution is found for too many iterations.

3.1.4 Procedure Max-Min Ant System for the SDWVRP

Algorithm 1 provides a complete algorithm based on the previous sections. We will refer to this algorithm specific as the MMAS_{SDWVRP}. In this algorithm one slight adjustment from Tang et al. (2013) is made, which is that the ants are not randomly distributed over n city nodes, because preliminary results showed that including this option to the algorithm reduced the solution quality.

Let t^{max} be the maximum iteration number and S_k the visited nodes by ant k in consecutive order. The objective function C is defined as in Section 2.1. C^t stands for the iteration-best found objective function value and C^* for the global best objective function value.

Algorithm 1: Roadmap MMAS_{SDWVRP}

Step 1: Initialization.

  $t = 0;\quad Q_k^{left} = Q\ \forall k,\quad C^t = C^* = \infty,\quad T_k = S_k = \emptyset\ \forall k;$

Step 2: Create an initial solution and initialize pheromone.

  Use a greedy algorithm (for example, choose in every step the closest next node, also with split deliveries) to create an initial solution;

  Set $\tau_{ij}(0) = \tau_{max}\ \forall (i,j)$ following (4);

Step 3: **if** $(t == t^{max})$ **then** Go to **Step 7**;

Step 4: Create m complete solutions.

for (*ant k = 1, ...m*) **do**

  Initialize S_k as $S_k = \{0\}$, such that ant k starts its first route from the depot;

  while (T_k *does not consist of all nodes*) **do**

    if $\left(V_c^k \setminus T_k \neq \emptyset \quad (q_j < Q_k^{left} \text{ and } q_j > 0)\right)$ **then**

      choose j from all possible nodes in $V_c^k \setminus T_k$ following (1) and (2);

      $T_k \leftarrow T_k + \{j\};\quad S_k \leftarrow S_k + \{j\};\quad q_j \leftarrow 0;\quad Q_k^{left} \leftarrow Q_k^{left} - q_j;$

      if $(Q_k^{left} == 0)$ **then** return to depot: $S_k \leftarrow S_k + \{0\};\quad Q_k^{left} \leftarrow Q;$

    else if $\left(V_c \setminus T_k \neq \emptyset \quad (q_j > 0)\right)$ **then**

      choose j from all possible nodes in $V_c \setminus T_k$ following (3) using split strategy;

      $q_j \leftarrow q_j - Q_k^{left};$ return to depot: $S_k \leftarrow S_k + \{0\};\quad Q_k^{left} \leftarrow Q;$

  Lower for the last route the series of Q_k^{left} for the not delivered load;

  Calculate the solution value C of ant k, update the iteration-best solution C^t;

Step 5: Update the global optimal solution: $C^* = \min(C^p),\quad p = \{0, 1, ..., t\};$

Step 6: Update the pheromone information.

  $t{+}{+}$, update $\tau_{ij}(t)\ \forall (i,j)$ following (6);

  if $(\tau_{ij}(t) < \tau_{min})$ **then** $\tau_{ij}(t) \leftarrow \tau_{min}$ following (5);

  if $(\tau_{ij}(t) > \tau_{max})$ **then** $\tau_{ij}(t) \leftarrow \tau_{max}$ following (4);

  Go to **Step 3**;

Step 7: Output the results, stop the process.

3.2 Max-Min Ant Systems for related Vehicle Routing Problems

So far, a Max-Min Ant System for the SDWVRP has been discussed. However, to evaluate whether this is the best model for a VRP with a weighted objective, some comparison with other models has to be made. To do so, we propose three different models which are very much related to MMAS$_{SDWVRP}$, but have some slight adjustments.

The first model is a MMAS for the Weighted Vehicle Routing Problem, denoted as MMAS$_{WVRP}$. In this model, the vehicle weight is taken into consideration, but no split deliveries will be performed. This implies that if an ant cannot serve a complete customer anymore, it will return to the depot to restore its capacity. After a solution is constructed, the load which was carried but unused during all routes is removed from the vehicle weight sequence, rather than only at the last route.

The second model is a MMAS for the Split Delivery Vehicle Routing Problem, denoted as MMAS$_{SDVRP}$. Here, split deliveries are included, but the vehicle weight is not considered. A different split delivery rule is used, where the split delivery node is chosen based on the minimization of the distance from the current node to the split delivery node plus the distance from the split delivery node back to the depot. The visibility parameter is redefined as $\eta_{ij} = 1/d_{ij}$. Solution qualities are compared using an objective function which minimizes only the total traveled distance of the ant's complete tour.

The third model is a MMAS for the Capacitated Vehicle Routing Problem, denoted as MMAS$_{CVRP}$. In this case, split deliveries are not performed and the vehicle weight is not taking into consideration. This model includes a combination of the restrictions as imposed by both the MMAS$_{WVRP}$ and the MMAS$_{SDVRP}$.

3.3 Candidate list

One way of trying to improve the solution quality is the addition of a candidate list, which has been done by Bell and McMullen (2004) and Rajappa et al. (2016). Using a candidate list, only a fixed ratio of the closest nodes surrounding that node can be visited. The idea of a candidate list is illustrated in Figure 2.

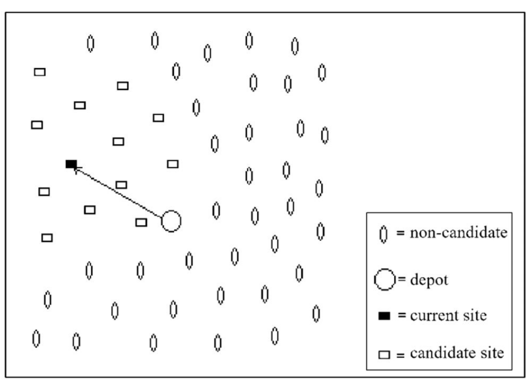


Figure 2: Visualization of a candidate list (Bell and McMullen 2004)

For the $MMAS_{SDWVRP}$, a full demand node is visited if it belongs to the (n/x) closest nodes surrounding the current position. Different values of x can be chosen to find the overall best neighborhood size. For instance, Rajappa et al. (2016) obtain the best results using $x = 9$.

If there is no node in the neighborhood of the current position for which the full demand can be satisfied, a split delivery is performed among the same (n/x) closest nodes. If there are no more node at all to visit in the neighborhood of the current position, the ant will return to the depot. This way, the ant possibly does not utilize its full load on a particular route. However, if an ant returns to the depot with remaining load, this load is always subtracted from the vehicle weight over the entire route. The depot node is a special case. Starting from there, the first node of the route is chosen according to the same candidate list structure. When there is no node to be visited left inside the depot candidate list, a node is chosen from the complete set of still to be visited nodes. This prevents that some nodes stay unvisited.

3.4 2-opt local search algorithm

Another way to improve the solution quality is to implement a 2-opt local search algorithm. Starting from an initial solution, a pair of nodes in a route are exchanged and the corresponding objective function for the exchanged solution is calculated (Dorigo and Stützle 2019). If an improvement is found, two nodes are exchanged and the algorithm continues, otherwise the algorithm is aborted. There is a distinction between the *first-improvement* rule, where the first exchange in a route which gives a better result is performed, and the *best-improvement* rule, where the best exchange in a route is chosen. The exchange of two nodes in a route is illustrated in Figure 3.

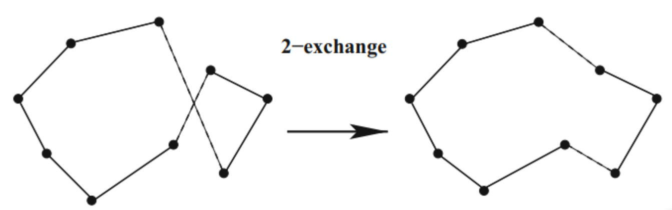


Figure 3: Visualization of a 2-exchange operation (Dorigo and Stützle 2019)

The algorithm could be performed on either the iteration-best solution or on the final solution only. An advantage of performing 2-opt on the iteration-best solution is that a good solution is obtained in relatively less iterations. However, this possibly causes early convergence to a local optimum when the ant colony optimization algorithm does not provide enough diversity in the search. An advantage of performing the 2-opt algorithm on the final solution only is that you are ensured to improve the final solution, or keep it the same. Although this is a benefit, this might not be as big of a cost reduction as would have been obtained with the 2-opt on every iteration-best solution.

Algorithm 2 shows the complete 2-opt procedure. Starting from a complete solution, every route from the tour is optimized by itself and all optimized routes are concatenated into a new tour. All possible exchanges for a single route are considered, but an exchange is only performed if this improves the solution, where this is either the first improvement or best improvement.

The vehicle loads are updated accordingly during the process. Further, `calculateObjective`(route) is the weighted objective function as described in Section 2.1 in case the algorithm is performed on the MMAS_{SDWVRP} or MMAS_{WVRP}. When 2-opt is performed on the MMAS_{SDVRP} or MMAS_{CVRP}, the `calculateObjective`(route) function is based on the total distance only, because this would be consistent with the fact that these models do not consider vehicle weight. A combination of the following settings will be tested: first-improvement rule plus 2-opt on every iteration best solution, best-improvement rule plus 2-opt on every iteration best solution and best-improvement rule plus 2-opt on the final solution only.

Algorithm 2: Roadmap 2-opt

```
   Initial: completeTour, a complete solution existing of n routes;
1  for (every route n (currentRoute) from completeTour) do
2      while (no improvement made) do
3          best_obj = calculateObjective(currentRoute);
4          for (i from first visited customer node up to and including second-to-last visited
             customer node) do
5              for (k from i+1 up to and including last visited customer node) do
6                  Create new route exchangedRoute where customer nodes i and k from
                     currentRoute are exchanged;
7                  new_obj = calculateObjective(exchangedRoute);
8                  if (new_obj < best_obj) then
9                      currentRoute = exchangedRoute;
10                     best_obj = new_obj;
11                     if (first-improvement rule) then Go to line 3;
12                     if (best-improvement rule) then Continue;
13 Concatenate all new routes and return the (possibly) improved solution;
```

4 Results

This section first describes the data and then provides a variety of computational results. All results are coded in Java and ran on a Windows 10, Intel i5 1.60 GHz Processor, 8 GB RAM laptop.

4.1 Data

The data used for this research is summarized in Table 1, which is also used in the research of Tang et al. (2013). This data is publicly available on the internet. The problem sizes vary from 16 to 401 nodes, including the depot. The corresponding vehicle capacities and mean and standard deviation of the weight distribution are presented as well.

Table 1: Summary statistics data

number	instance name	num. nodes	capacity veh.	avg. weight	stdev. weight
P01	A-n33-k6	33	100	16.9	11.21
P02	A-n65-k9	65	100	13.7	7.28
P03	E-n76-k7	76	220	18.2	7.91
P04	B-n41-k6	41	100	14.2	5.59
P05	B-n63-k10	63	100	14.9	8.22
P06	E-n30-k3	30	4500	439.7	600.02
P07	F-n45-k4	45	2010	164.1	255.59
P08	ulysses-n16-k3	16	5	1	0
P09	att-n48-k4	48	15	1	0
P10	F-n135-k7	135	2210	109.1	186.61
P11	ta-n101-k11b	101	1842	195.0	263.09
P12	F-n72-k4	72	30,000	1617.5	29,867.79
P13	kelly01	241	550	20	10
P14	kelly02	321	700	20	10
P15	kelly03	401	900	20	10

4.2 Computational results

Table 2 shows the results of the MMAS_{SDWVRP}, MMAS_{WVRP}, MMAS_{SDVRP} and MMAS_{CVRP} models as presented in Sections 3.1 and 3.2, performed on the fifteen data instances. All reported objective function values are from the weighted objective function. For MMAS_{SDVRP} and MMAS_{CVRP} in particular, it means that these models create a solution based on the shortest distance only, but are evaluated in a weighted objective function.

The following parameter settings are used: $q_0 = 0.5$, $\alpha = 1$, $\beta = 2$, $\rho = 0.98$. These settings are taken from Stützle and Hoos (2000) and slightly adjusted after parameter tuning. Further, the number of ants equals the number of nodes, the maximum number of iterations is 1000 and the vehicle weight is assumed to be half the capacity of the vehicle. The global best solution updates the pheromone trails instead of the local best solution, because this typically resulted in better solutions.

Table 2: Computational results different MMAS models calculated with a weighted objective and results MMAS-SD by Tang et al. (2013)

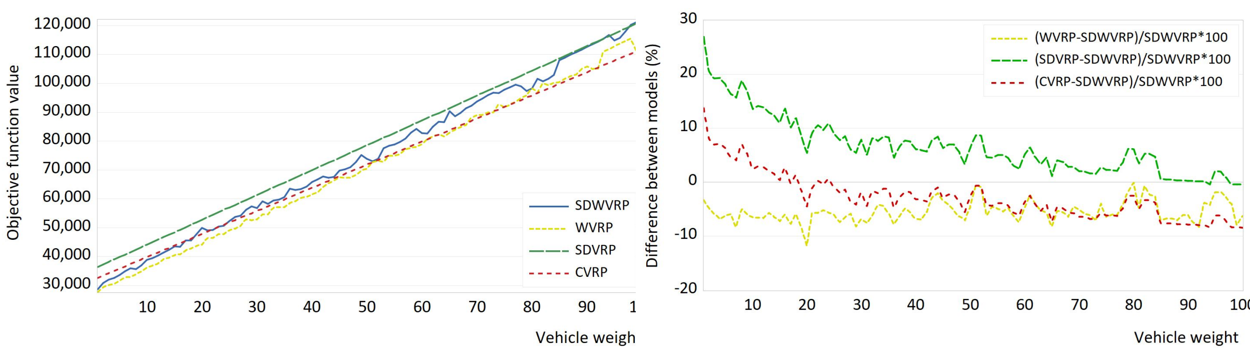
	MMAS_{SDWVRP}	MMAS_{WVRP}	MMAS_{SDVRP}	MMAS_{CVRP}	Tang et al. (2013)
P01	73,840	**70,382**	78,531	71,799	81,532
	(2.7 s)	(2.9 s)	(2.9 s)	(2.5 s)	(3.9 s)
P02	**120,418**	124,760	120,951	124,727	111,007
	(17.3 s)	(16.3 s)	(15.6 s)	(15.5 s)	(34 s)
P03	157,552	**150,950**	192,137	160,977	188,555
	(26.1 s)	(26.0 s)	(26.7 s)	(26.4 s)	(43 s)
P04	83,437	**80,607**	81,141	82,846	136,049
	(6.0 s)	(5.3 s)	(5.7 s)	(5.0 s)	(7.0 s)
P05	156,578	156,255	**155,613**	156,437	167,127
	(15.5 s)	(14.9 s)	(14.5 s)	(15.4 s)	(30 s)
P06	2,219,333	**2,204,273**	2,464,892	2,524,723	2,505,084
	(2.4 s)	(2.4 s)	(2.4 s)	(2.0 s)	(3.3 s)
P07	1,245,667	**1,244,737**	1,369,424	1,301,145	1,534,857
	(6.2 s)	(5.7 s)	(7.0 s)	(5.8 s)	(7.8 s)
P08	**386**	**386**	433	433	365
	(0.4 s)	(0.4 s)	(0.4 s)	(0.4 s)	(0.4 s)
P09	**566,237**	**566,237**	621,989	621,989	584,290
	(7.9 s)	(7.6 s)	(7.9 s)	(8.2 s)	(10 s)
P10	2,902,025	2,919,124	2,891,070	**2,818,971**	2,388,835
	(141.2 s)	(140.7 s)	(140.5 s)	(141.1 s)	(49 s)
P11	4,229,880	**4,002,839**	4,569,486	4,339,438	4,518,946
	(49.9 s)	(50.3 s)	(53.1 s)	(49.9 s)	(119 s)
P12	8,044,020	8,317,233	8,483,870	**7,374,379**	10,331,621
	(20.3 s)	(19.0 s)	(22.1 s)	(22.4 s)	(362 s)
P13	**3,194,354**	3,291,377	3,291,377	**3,194,354**	3,669,419
	(843.2 s)	(842.6 s)	(836.8 s)	(839.1 s)	(475 s)
P14	**6,200,207**	**6,200,207**	6,661,727	6,514,039	6,878,197
	(2033.9 s)	(2068.0 s)	(1965.6 s)	(1993.5 s)	(935 s)
P15	**10,199,214**	**10,199,214**	11,349,380	11,349,380	11,525,875
	(3940.9 s)	(3832.2 s)	(3891.0 s)	(3887.7 s)	(2364 s)

The first observation from Table 2 is that the results of MMAS_{SDWVRP} are in range with the results of the SDWVRP solutions found by Tang et al. (2013) with MMAS-SD, but not consistently better or worse than their solutions. This is as expected, because of the random nature of this heuristic and the missing parameter settings by Tang et al. (2013), the vehicle weight in particular. The running times we obtained are shorter for small instances and larger for big instances, but still fairly consistent with their running times. Also, the differences in running times between different MMAS models are too small to have a share in the decision which MMAS model to use.

The second observation from Table 2 is that MMAS_{WVRP} 10 times found the best solution, followed by MMAS_{SDWVRP} with 6 best solutions, then $MMAS_{CVRP}$ with 3 best solutions and MMAS_{SDVRP} with only 1 best solution. This means that the MMAS algorithm for the SDWVRP is not the superior algorithm in this case. One possible reason why the model without split deliveries performs better than the one with split deliveries has to do with the lower vehicle weight without split deliveries. Because the MMAS$_{WVRP}$ never performs a split delivery, it possibly performs more routes where a lower initial vehicle load is needed. This implies that the vehicle weight is generally lower on more paths, which has a big impact on the weighted objective function value. However, this might be a too strong assumption: if we calculate the objective purely based on the traveled distances for the first 9 out of 10 instances for which split deliveries are possible (P01-P07 and P10-P12), the better algorithm both had the lowest distance-based as well as weighted objective function, comparing MMAS$_{SDWVRP}$ and MMAS$_{WVRP}$. Only for instance P10 the MMAS$_{WVRP}$ had a slightly lower distance-based objective than MMAS$_{SDWVRP}$, but the split deliveries variant still had a better weighted objective. This relationship will be further investigated in Section 4.2.4.

4.2.1 Varying vehicle weight

So far we assumed that the fixed vehicle weight is equal to half the capacity. However, the mutual results between the different MMAS algorithms differ for different assumed vehicle weights. To illustrate this, Figure 4a shows the results for MMAS$_{SDWVRP}$, MMAS$_{WVRP}$, MMAS$_{SDVRP}$ and MMAS$_{CVRP}$ for data instance P01 with a vehicle weight on the interval $\{1, 2, \ldots, 100\}$, which is between one hundredth part of the vehicle capacity and the full capacity. Figure 4b shows the objective function difference between MMAS$_{SDWVRP}$ and the other three models. Data points below the zero line are the results better than the MMAS$_{SDWVRP}$ results.



(a) Different objective function values

(b) Different models compared to MMAS$_{SDWVRP}$

Figure 4: Varying vehicle weight P01

In Figure 4a, the MMAS$_{SDVRP}$ and MMAS$_{CVRP}$ results are straight lines, because it returns the same tour independent of the vehicle weight. For a low vehicle weight, the MMAS$_{SDVRP}$ results are much higher than for the MMAS$_{SDWVRP}$, but this difference gets smaller the higher the vehicle weight becomes. A similar relationship holds for the MMAS$_{CVRP}$ results. The MMAS$_{WVRP}$ is more or less constantly better than the MMAS$_{SDWVRP}$. In general, the MMAS$_{SDWVRP}$ results are less powerful when the assumed vehicle weight is large. The observation that split deliveries are less powerful at a high vehicle weight might be explained by the fact that a vehicle has to carry relatively much weight on a longer distance if a vehicle travels back to the the depot after a split delivery.

We have seen this in the example in Section 2.2 as well: if a higher vehicle weight is assumed, the solution with a split delivery gets worse compared to the solution without split deliveries. Next to that, a reason why the MMAS$_{SDVRP}$ results improve compared to the MMAS$_{SDWVRP}$ at a higher vehicle weight is that the fixed vehicle weight starts to dominate the variable cargo weight, which makes the traveled distances relatively more important compared to the cargo weight. Both could explain the lower quality solutions of MMAS$_{SDWVRP}$ in comparison with other models in Table 2.

4.2.2 Results candidate list

In Figure 5, the results for candidate list sizes $\{n/2, n/3, ..., n/14\}$ are given for data instances P01-P07 and P09-P12. In Figure 5, the number of nodes per instance are displayed directly next to the lines to see the different trends of cost savings for different instance sizes. The results of P08 are omitted, because the candidate list resulted in high objective function increase for this instance. It is likely that the results for P08 were poor because P08 has only 15 customer nodes, which results in too little possibilities to choose from when applying a candidate list. For all candidate list sizes, the ceiling function is applied on the particular size.

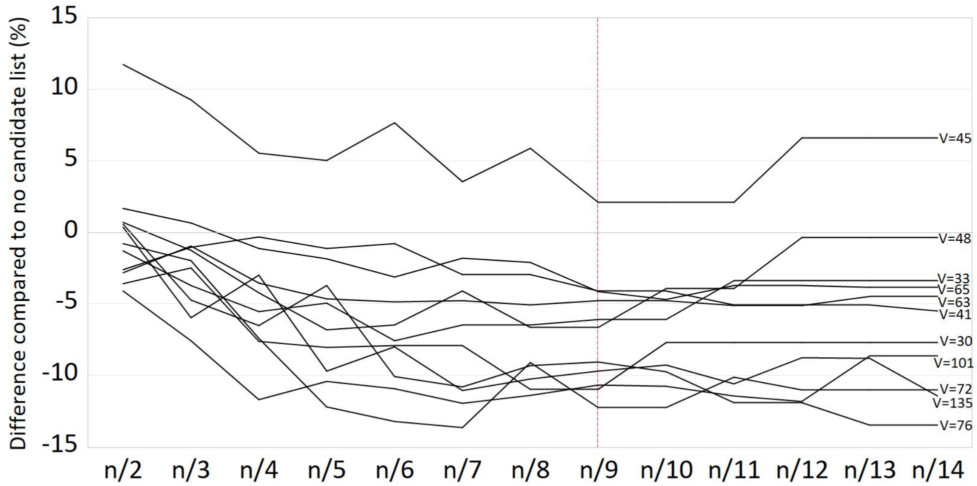


Figure 5: Results of different candidate list sizes for instance P01-P07 and P09-P12

As becomes clear from Figure 5, all data instances take advantage of a candidate list, except P07. On the right side of the graph more straight lines are visible, because more candidate list sizes collide. For example, applying a candidate list $n/11$ up until $n/14$ on 33 nodes all result in a candidate list size of 3. The average best candidate list over the eleven tested instances is $n/9$. Surprisingly, this is the same best candidate list size as found by Rajappa et al. (2016), who perform it on the SDVRP. The $n/9$ candidate list implementation lowers the objective function value on average with 4.5%, and even with 6.9% if P08 is omitted. Furthermore, it seems that in general a candidate list has a bigger effect on the instances with many nodes than the ones with fewer nodes. A possible explanation why a candidate list works well for the SDWVRP is that split deliveries are only performed 'in the neighborhood' of a node. This means that split deliveries are only performed when they are more likely to be worthwhile. And is some cases, more small routes with a low average vehicle weight might be more beneficial than fewer long routes with high average vehicle weight. In particular, the MMAS algorithm is designed to always serve nodes with a low demand if the ant still has some capacity left, even if these node are far away from the current position. This is prevented when applying a candidate list. Considering the running time, with our implementation it takes at least a candidate list of $n/8$ to halve the running time when applying on a bigger instance, like P10 with 135 nodes.

One last result is that in general applying a candidate list for the depot node if possible works better than no candidate list for the depot node. This might be explained by the fact that an ant always starts with a high load leaving from the depot. Therefore, it has a big influence on the weighted objective function if the ant has to travel from the depot to a node far way from the depot.

4.2.3 Results 2-opt local search

The 2-opt algorithm with best-improvement rule was applied on every iteration-best solution before updating the global-best solution for the first 12 data instances (P01-P12). The results are shown in Figure 6. The solutions improved in 10 out of 12 cases, with an average improvement of 2.7%. For P09 the solution was slightly worse and for P08 the solution stayed the same. The 2-opt local search heuristic performed better on larger instances, generally speaking. The running time was only about 4% higher for the biggest instance.

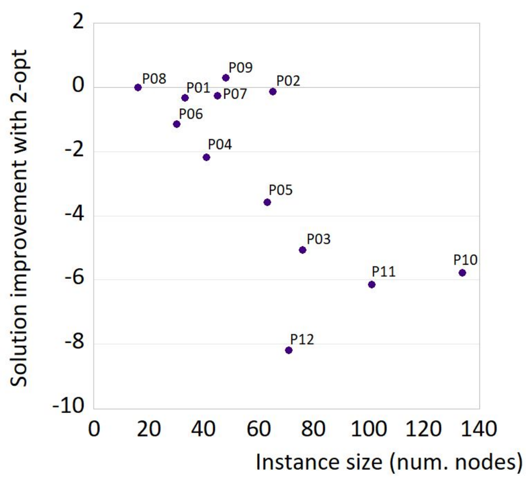


Figure 6: Results of 2-opt with best-improvement rule on every iteration-best solution

The same algorithm was also performed in two different versions: the first one with the first-improvement rule and iteration-best solution and the second one with the best-improvement rule, but only on the final solution. Both versions turned out to perform worse on average over the twelve instances, they only reduced the objective function value by 1.7% and 1.8% on average.

Comparing with results obtained by a $n/9$ candidate list, the candidate list gave a better result than the 2-opt algorithm in 10 out of 12 cases. The other two cases are instance P07 and P08 where the candidate list did not improve the solution and 2-opt only found a small improvement or an equivalent solution. This could be an indication that the original MMAS_{SDWVRP} solution was already close to the optimal solution, which is why a candidate list did not work for these instances. In conclusion, computational results show that using a candidate list of $n/9$ is more beneficial than a 2-opt algorithm for the majority of the problem instances.

4.2.4 Results candidate list and 2-opt local search combined

Combining both a candidate list of $n/9$ and a 2-opt algorithm yields the best results for the MMAS_{SDWVRP}. An exception is P08: this one never improves its original solution. One interesting observation is that for instances under 65 nodes a 2-opt algorithm only performed on the final solution works best, where this is a 2-opt algorithm performed on every iteration-best solution for instances over 65 nodes. Strategy A is set up to obtain improved solutions for different MMAS models.

Strategy A: Settings to improve MMAS solutions

1: Apply the original $\text{MMAS}_{(.)VRP}$ algorithm to instances with less than 20 nodes;

2: Apply a candidate list of size $n/9$ in combination with a 2-opt algorithm on the final solution to instances between 20 and 65 nodes;

3: Apply a candidate list of size $n/9$ in combination with a 2-opt on the iteration-best solution to instances with more than 65 nodes.

Finally, we perform Strategy A once again on all instances for the four different MMAS algorithms. Table 3 presents improved solutions compared to Table 2, labeled as `new`. For $\text{MMAS}^{\text{new}}_{SDWVRP}$, the improvement with MMAS_{SDWVRP} is listed as well, denoted as Improv. For convenience, running times are omitted.

Table 3: Computional results of improved MMAS models with candidate list and 2-opt algorithm

	$\text{MMAS}^{\texttt{new}}_{SDWVRP}$	Improv.	$\text{MMAS}^{\texttt{new}}_{WVRP}$	$\text{MMAS}^{\texttt{new}}_{SDVRP}$	$\text{MMAS}^{\texttt{new}}_{CVRP}$
P01	**67,697**	(-8.32%)	67,722	74,808	74,036
P02	**110,854**	(-7.94%)	112,304	114,430	113,163
P03	**138,670**	(-11.98%)	140,888	160,983	152,710
P04	79,359	(-4.89%)	**78,859**	80,612	81,043
P05	145,246	(-7.24%)	**144,465**	151,955	148,590
P06	**1,907,453**	(-14.05%)	1,932,024	2,025,544	2,002,292
P07	1,167,157	(-6.30%)	**1,150,102**	1,169,542	1,169,542
P08	**386**	(-0.00%)	**386**	433	433
P09	**521,783**	(-7.85%)	**521,783**	595,544	595,544
P10	**2,504,083**	(-13.71%)	2,562,431	2,576,535	2,651,436
P11	**3,536,227**	(-16.40%)	3,542,782	3,732,894	3,667,378
P12	**6,659,449**	(-17.21%)	6,698,987	6,743,824	7,062,737
P13	2,871,860	(-10.10%)	**2,771,685**	3,121,938	3,143,805
P14	**5,703,066**	(-8.02%)	**5,703,066**	5,993,004	6,052,258
P15	**8,889,052**	(-12.85%)	8,988,791	10,453,052	9,993,405

First we remark that for $\text{MMAS}^{\text{new}}_{SDVRP}$ and $\text{MMAS}^{\text{new}}_{CVRP}$ a 2-opt algorithm is used which compares solutions based on a distance-based objective, as mentioned in Section 3.4. This possibly worsens the weighted objective function value compared to not applying the 2-opt algorithm. Performing a weighted 2-opt algorithm on these solutions generally improves the solutions, and even in some cases the results become better than the results of $\text{MMAS}^{\text{new}}_{SDWVRP}$ and $\text{MMAS}^{\text{new}}_{WVRP}$.

Table 3 shows that a $n/9$ candidate list in combination with a 2-opt algorithm greatly improves the weighted objective function values from Table 2, where $\text{MMAS}^{\text{new}}_{SDWVRP}$ improved on average with 9.8% compared with MMAS_{SDWVRP}. The $\text{MMAS}^{\text{new}}_{SDWVRP}$ is now slightly better than the $\text{MMAS}^{\text{new}}_{WVRP}$, with a mean difference of 0.2%. The results of the non-weighted models $\text{MMAS}^{\texttt{new}}_{SDVRP}$ and $\text{MMAS}^{\texttt{new}}_{CVRP}$ are generally improved, but the difference with the the weighted models get bigger.

With a candidate list, the number of routes generally increases, but the total traveled length decreases. By clustering the nodes, more efficient routes are created. In general, it seems that the solution quality is a combination of the series of vehicle weights along the routes, the number of routes and the number of split deliveries. More specific results for instances P01-P07 and P10-P12 are shown in Table 4. For 6 out of 10 instances, the best model for the weighted objective also has the lowest distance objective. MMAS_{SDWVRP} has a lower number of routes for 3 instances, where this is 1 time the case for MMAS_{WVRP}. Lastly, Table 4 does not show a strong general pattern between the number of split deliveries and the best model for the weighted objective.

Table 4: More specific results for the results of $\text{MMAS}^{new}_{SDWVRP}$ and MMAS^{new}_{WVRP}

Instance	$\text{MMAS}^{new}_{SDWVRP}$			MMAS^{new}_{WVRP}		Best $\text{MMAS}^{new}_{(\cdot)VRP}$ model
	num. splits	num. routes	distance obj.	num. routes	distance obj.	weighted obj.
P01	1	6	**773.17**	7	821.96	SDW
P02	5	9	**1227.00**	10	1267.94	SDW
P03	2	8	784.24	7	**772.41**	SDW
P04	1	8	937.33	8	**933.21**	W
P05	3	11	**1601.58**	11	1608.56	W
P06	0	4	**512.33**	4	528.45	SDW
P07	0	6	832.29	6	**815.53**	W
P10	3	8	**1394.65**	9	1431.44	SDW
P11	3	13	2322.72	13	**2320.12**	SDW
P12	0	6	**308.00**	6	314.33	SDW

In conclusion, split deliveries in the $\text{MMAS}^{\texttt{new}}_{SDVWRP}$ help to reduce the objective function value further, but this is not automatically the case. Follow-up studies of heuristics for the SDWVRP therefore might come up with a algorithm which only performs split deliveries if they are useful, instead of always performing a split delivery if possible. Another interesting topic of further research could be the inclusion of a local update of pheromone τ_{ij} after every path that is traveled. Some first results were promising, but we did not find general settings which works well on most data instances. Mazzeo and Loiseau (2004) could be used to test different local update rules.

5 Conclusion

This thesis focuses on the Max-Min Ant System for the Split Delivery Weighted Vehicle Routing Problem, as introduced by Tang et al. (2013). The goal of this research is to revise the computational results by Tang et al. (2013) and compare them with three simplified versions of the MMAS for the related WVRP, SDVRP and CVRP. Moreover, the current algorithm is extended with a candidate list and 2-opt local search algorithm to investigate whether results could be improved.

The first outcome is that MMAS_{WVRP} was found to be about 1% better than MMAS_{SDWVRP}, without extensions. The MMAS_{SDVRP} and MMAS_{CVRP} were generally not better, but found the best result in respectively 3 and 1 out of 15 cases. These results are in contradiction with the conclusion of Tang et al. (2013) that the SDWVRP is superior relative to the WVRP. However, which algorithm performs best depends on the assumed vehicle weight, amongst others. First test results found that the higher the assumed vehicle weight, the less powerful the MMAS_{SDWVRP} is compared to its counterparts.

The second outcome is that a candidate list of size $n/9$ improves the solutions the most on average and has a stronger effect on the bigger data instances. A candidate list on itself turned out to achieve higher cost reductions than the 2-opt algorithm. Nevertheless, combining the $n/9$ candidate list and the 2-opt algorithm found a cost reduction in 14 out of 15 cases for the MMAS_{SDWVRP}, for which an average reduction of 9.8% was found. An exception was made for the smallest instance with 16 nodes, which did not improve using a candidate list and a 2-opt algorithm. With these two extensions, the MMAS_{SDWVRP} has slightly better results than the MMAS_{WVRP}, but the differences are small. Comparing these models, split deliveries seem to reduce the average traveled distance, but no strong relationship has been found.

Finally, we propose some topics for further research. First of all, the current MMAS-SD as proposed by Tang et al. (2013) always performs a split delivery if possible. However, computational results have shown that a MMAS algorithm without split deliveries is not inferior to the one with split deliveries. One might come up with a model which only performs split deliveries when they are beneficial, or at least does not always require a split delivery when one is possible. A first step in doing so has been taken by including a candidate list. A second valuable topic of research is the inclusion of a local pheromone update rule for the MMAS. First results were promising, but further research needs to be done to determine the best local update rules for the SDWVRP. One last field of research could be to compare the extended MMAS_{SDWVRP} to other heuristics especially created for the CVRP, WVRP and SDVRP, or an optimal solution if possible. This has been done by Tang et al. (2013), but to our best understanding not with a candidate list and a 2-opt heuristic.

References

Archetti, C., M.W.P. Savelsbergh, and M.G. Speranza. “Worst-Case Analysis for Split Delivery Vehicle Routing Problems”. In: *Transportation Science* 40.2 (2006), pp. 226–234.

Archetti, C., M.W.P. Savelsbergh, and M.G. Speranza. “To split or not to split That is the question”. In: *Transportation Research Part E: Logistics and Transportation Review* 44.1 (2008), pp. 114–123.

Archetti, C. and M.G. Speranza. “Vehicle routing problems with split deliveries”. In: *Intl. Trans. in Op. Res.* 19 (2012), pp. 3–22.

Bektaş, T. and G. Laporte. “The Pollution-Routing Problem”. In: *Transportation Research Part B: Methodological* 45.8 (2011), pp. 1232–1250.

Bell, J.E. and P.R. McMullen. “Ant colony optimization techniques for the vehicle routing problem”. In: *Advanced Engineering Informatics* 18.1 (2004), pp. 41–48.

Bullnheimer, B., R.F. Hartl, and C. Strauss. “Applying the ANT System to the Vehicle Routing Problem”. In: *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*. Boston, MA: Springer US, 1999, pp. 285–296.

Camm, J.D., M.J. Magazine, S. Kuppusamy, and K. Martin. “The demand weighted vehicle routing problem”. In: *European Journal of Operational Research* 262.1 (2017), pp. 151–162.

Dantzig, G.B. and J.H. Ramser. “The Truck Dispatching Problem”. In: *Management Science* 6.1 (1959), pp. 80–91.

Dorigo, M. and L.M. Gambardella. “Ant colony system: a cooperative learning approach to the traveling salesman problem”. In: *IEEE Transactions on Evolutionary Computation* 1.1 (1997), pp. 53–66.

Dorigo, M., V. Maniezzo, and A. Colorni. *The Ant System: an autocatalytic optimizing process*. Tech. rep. Dipartimento di Elettronica e Informazione Politecnico di Milano, 1991.

Dorigo, M. and T. Stützle. “Ant Colony Optimization: Overview and Recent Advances”. In: *Handbook of Metaheuristics*. Cham: Springer International Publishing, 2019, pp. 311–351.

Dror, M. and P. Trudeau. “Savings by Split Delivery Routing”. In: *Transportation Science* 23.2 (1989), pp. 141–145.

Golden, B.L., S. Raghavan, and E.A. Wasil. *The vehicle routing problem: latest advances and new challenges*. Springer Science Business Media, 2010.

Mazzeo, S. and I. Loiseau. “An Ant Colony Algorithm for the Capacitated Vehicle Routing”. In: *Electronic Notes in Discrete Mathematics* 18 (2004), pp. 181–186.

Rajappa, G.P., J.H. Wilck, and J.E. Bell. “An Ant Colony Optimization and Hybrid Metaheuristics Algorithm to Solve the Split Delivery Vehicle Routing Problem.” In: *International Journal of Applied Industrial Engineering (IJAIE)* 3.1 (2016), pp. 55–73.

Stützle, T. and H.H. Hoos. “MAX-MIN Ant System and local search for the traveling salesman problem”. In: *Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC ’97)*. 1997, pp. 309–314.

Stützle, T. and H.H. Hoos. “MAX–MIN Ant System”. In: *Future Generation Computer Systems* 16.8 (2000), pp. 889–914.

Tang, J., Y. Ma, J. Guan, and C. Yan. “A Max–Min Ant System for the split delivery weighted vehicle routing problem”. In: *Expert Systems with Applications* 40.18 (2013), pp. 7468–7477. DOI: `https://doi.org/10.1016/j.eswa.2013.06.068`.

Zhang, Z., H. Qin, W. Zhu, and A. Lim. “The single vehicle routing problem with toll-by-weight scheme: A branch-and-bound approach”. In: *European Journal of Operational Research* 220.2 (2012), pp. 295–304.