The Multi-Compartment Multi-Vehicle Routing Problem

Martijn Nieuwenhuis 348488

February 14, 2018

Supervisor: T.A.B. Dollevoet Second Assessor: T. Breugem

Erasmus University Rotterdam
Erasmus School of Economics
Bachelor Thesis Econometrics and Operational Research

Abstract

The multi-compartment vehicle routing problem (MCVRP) frequently arises in practical applications where different products need to be handled or kept differently. Where the MCVRP uses a fleet of identical vehicles, this research uses a non-homogeneous fleet of vehicles. This extends the MCVRP to the multi-compartment multi-vehicle routing problem (MCMVRP). We get solutions for both the MCVRP and the MCMVRP by means of an iterated tabu search algorithm proposed by Silvestrin and Ritt (2017). We find that in general the non-homogeneous fleet composition does not find better solutions than a fleet of identical vehicles.

Contents

1 Introduction

This research is based on the article 'An iterated tabu search for the multi-compartment vehicle routing problem' by Silvestrin and Ritt (2017). Since this paper only considers identical vehicles, this research will look into the possibility to use different type of vehicles. Therefore, the following question is the base of the research:

Can the solution of the multi-compartment vehicle routing problem be improved by using a non-homogeneous fleet of vehicles?

The problem that needs to be solved is a variation of the well known vehicle routing problem (VRP), where a fleet of identical vehicles with a certain capacity is used to serve a set of customers that each have a different demand for a certain product, while minimizing the number of vehicles used or the total time needed to serve all customers. This variation involves the availability of different types of products, that each need to be handled differently. Therefore, the vehicles must have different compartments to separate the products, which expands the VRP to the multi-compartment vehicle routing problem (MCVRP). We see a lot of different applications of this problem in practice. For instance, when some of the transported products need to be refrigerated, the vehicles must have some sort of cooling compartment. Another example where different compartments are considered is in livestock transportation, to separate animals of different species or characteristics.

The fleet of the MCVRP consists of identical vehicles with a predetermined capacity for each type of product. This research however expands the fleet to multiple vehicle types such that the fleet is non-homogeneous. The resulting problem is called the multi-compartment multi-vehicle routing problem (MCMVRP). The use of vehicles with different divisions of the compartments could be preferred to the use of identical vehicles when there is for instance large variation in the demand of the products per region. In case of the livestock problem, one region might have farms with more chicken than pigs, where this ratio is vice versa in another region. Then the vehicles which are preferred to be used in these regions also represent this ratio in their compartment division. Therefore the use of different type of vehicles will probably be preferred compared to using identical vehicles.

Silvestrin and Ritt (2017) showed that their iterated tabu search (ITS) generally outperformed previously suggested algorithms for the MCVRP in solution quality and in computation time. This research therefore also implements their ITS algorithm, although slight adjustments are made to extend the MCVRP to the MCMVRP to account for the change of the fleet.

The research uses some of the instances provided by Christofides et al. (1979). These CMT instances contain locations and the demand of 50 to 120 customers. Since these instances are generated for the VRP, the given demand of each customer is divided such that it has a demand for each of the available products.

In the next section, we describe the MCMVRP formally, present the mathematical model and explain the fleet composition. The research methodology is discussed in section 3 where we discuss the details of the ITS. In section 4 the instances are described and we summarize the results of the ITS on both the MCVRP as the MCMVRP. Finally we give the conclusions of this research in section 5.

2 Problem Description

The MCMVRP problem is a slight variation of the MCVRP problem proposed by Silvestrin & Ritt (2017) which uses a fleet of identical vehicles with multiple compartments. The fleet of the MCMVRP however is non-homogeneous, in the sense that the fleet consists of different type of vehicles with varying capacity for each product.

A set of locations is given by $V = \{V_0\} \cup V_+$, where V_0 is the depot and V_+ is the set of n customers. The travel time between each pair of locations $i, j \in V$ is given by d_{ij}. Since this travel time is equal to the euclidean distance between the locations, we assume symmetric travel time: $d_{ij} = d_{ji}$. The set of available products is given by $P = [m]$, where we use notation $[m] = \{1, ..., m\}$. Each customer $i \in V_+$ has a demand for each product $p \in P$ given as c_{ip}. The fleet of vehicles $F = [f]$ consist of f vehicles with capacity C_p^k, where $k \in F, p \in P$. The division of the capacity of the vehicles will be explained in section 2.3. A valid route of a vehicle starts at the depot, visits a number of customers, and returns to the depot. Each time a vehicle stops at a customer site, it satisfies at least one of its demands. Therefore, a customer may be on different routes. A visit v is defined as a pair of a customer $V(v)$ and a set of products $P(v)$ delivered at this visit: $v = (V(v), P(v))$. A route is represented by an ordered subset $R = \{v_1, ..., v_{l(R)}\}$, with $l(R)$ equal to the total number of visits in the route. The set of visited customers in a route is then given by $V(R) = \{V(v) | v \in R\}$ and the set of attended customer-demand pairs $P(R) = \{(V_i, p) | V_i \in v, p \in P, (V_i, p) \in R\}$. The total time $d(R)$ and demand $c(R)$ of a route are calculated as

$$d(R) = d_{V_0, V(v_1)} + \sum_{1 \leq i < l(R)} d_{V(v_i), V(v_{i+1})} + d_{V(v_{l(R)}), V_0}$$

$$c(R) = \sum_{p \in P} c(R)_p$$

$$c(R)_p = \sum_{1 \leq i \leq l(R) | p \in P(v_i)} c_{V(v_i), p}$$

We want to find a set of valid routes $S = \{R_1, ..., R_f\}$ such that a customer-demand pair is only on one route, while satisfying all demand, without exceeding the capacity constraints for which the total time is minimized.

2.1 Mathematical Model

The MCMVRP can be modeled as an integer linear problem the same way as the mathematical model of the MCVRP by Silvestrin and Ritt (2017) where we adjust the capacity constraints, such that different types of vehicles are allowed. We introduce the following binary decision variables:

x_{ijk} := vehicle k visits arc (i, j),
y_{jkp} := the demand of client j for product p is attended by vehicle k,

$$
\begin{aligned}
\textbf{min.} & \sum_{i,j \in V} \sum_{k \in F} d_{ij} x_{ijk} & & (1)\\
\textbf{s.t.} & \sum_{i \in V} x_{ijk} \leq 1, & \forall j \in V_+, \forall k \in F, & \quad (2)\\
& \sum_{i \in V} x_{ijk} = \sum_{i \in V} x_{jik}, & \forall j \in V, \forall k \in F, & \quad (3)\\
& \sum_{ij \in S} x_{ijk} \leq |S| - 1, & \forall S \subseteq V_+, |S| \geq 2, \forall k \in F, & \quad (4)\\
& y_{jkp} \leq \sum_{i \in V} x_{ijk}, & \forall j \in V_+, \forall k \in F, \forall p \in P, & \quad (5)\\
& \sum_{k \in F} y_{jkp} = 1, & \forall j \in V_+, \forall p \in P, & \quad (6)\\
& \sum_{j \in V_+} c_{jp} y_{jkp} \leq C_p^k & \forall k \in F, \forall p \in P, & \quad (7)\\
& x_{ijk} \in \{0, 1\} & \forall i, j \in V, \forall k \in F, & \quad (8)\\
& y_{jkp} \in \{0, 1\} & \forall i, j \in V_+, \forall k \in F, \forall p \in P, & \quad (9)
\end{aligned}
$$

The total traveled time is minimized in (1). Constraint (2) makes sure that a customer is visited at most once on a route. The next constraint (3) ensures route flow conversation. There are no sub routes allowed due to constraint (4), and constraint (5) links the decision variables to each other. Constraints (6) realize that the demand of a customer for a product is attended by one visit. The capacity constraints are given by (7).

In order to gain more insight in constraint (4), where sub routing is prevented, we look at a set of clients $U \subseteq V^+$, where the clients are on the same route. If we want to create the largest possible sub route, we would need $|U|$ arcs (thus the left hand side of constraint (4) equals to $|U|$). In this case the creation of this largest sub route in set U is prevented since constraint (4) (with $S = U$) allows only routes with at most $|U| - 1$ arcs. If we make a smaller sub route of the clients in set $|U|$, for example between the clients in set $W \subset U$, then it is seen that this sub route is allowed in constraint (4) (with $S = U$) since $|W| \leq |U| - 1$. Although, constraint (4) (with $S = W$) prevents the creation of the sub route of set W in the same way that is showed for set U. Therefore, since constraint (4) applies for all combinations of set $S \subseteq V^+$, all constraints (4) combined prevent the creation of sub routes.

2.2 Fleet composition

Fleet of the MCVRP

Before explaining the iterated tabu search, we first describe the composition of the fleet of vehicles. As mentioned before, the MCVRP uses a fleet of identical vehicles with m compartments. Therefore, $C_p^i = C_p^j$ for all $i, j \in F, p \in P$. We only consider two different products, $p \in \{1, 2\}$, thus the capacity of each compartment of the MCVRP vehicles is decided as follows:

$$C_p^k = C * \frac{\overline{D}_p}{\overline{D}_1 + \overline{D}_2}, \text{ for } p \in P, k \in F,$$

where $\overline{D}_p$ is the average demand of product p and C is the total capacity of a vehicle. This ensures that the ratio between the average of the different products is also represented in the vehicle's capacity.

Fleet of the MCMVRP

The fleet of the MCMVRP on the other hand is non-homogeneous, where we use two different types of vehicles. We therefore introduce two new sets, $T_1 \subset F$ and $T_2 \subset F$, representing the vehicles of type 1 and type 2 respectively. Obviously, a vehicle can only be of one type: $T_1 \cap T_2 = \emptyset$. The first type has slightly more capacity for the first product than the vehicles in the MCVRP, but less capacity for the second product. The capacity of the second type differs the other way around: more capacity for the second product, but less capacity for the first product. We choose $\beta \in [0, \min_{p \in P} \frac{\overline{D}_p}{\overline{D}_1 + \overline{D}_2}]$, as the wanted deviation in the vehicles' capacity opposed to the average demand proportion. If β is chosen as $\min_{p \in P} \frac{\overline{D}_p}{\overline{D}_1 + \overline{D}_2}$ we get that either of the two compartments of one of the vehicles is empty, and therefore β cannot exceed this. We get the following vehicle capacities:

$$C_1^i = C * (\frac{\overline{D}_1}{\overline{D}_1 + \overline{D}_2} + \beta) \text{ for } i \in T_1,$$
$$C_2^i = C * (\frac{\overline{D}_2}{\overline{D}_1 + \overline{D}_2} - \beta) \text{ for } i \in T_1,$$
$$C_1^j = C * (\frac{\overline{D}_1}{\overline{D}_1 + \overline{D}_2} - \beta) \text{ for } j \in T_2,$$
$$C_2^j = C * (\frac{\overline{D}_2}{\overline{D}_1 + \overline{D}_2} + \beta) \text{ for } j \in T_2,$$

Note that the fleet reduces to the identical vehicles of the MCVRP for $\beta = 0$.

Finally, we want the fleet to represent the ratio in the demand as closely as possible, and therefore we let the fleet of vehicles consist of the same number of vehicles of either type: $|T_1| = |T_2|$. If the number of vehicles is odd, then we choose that vehicle type T_p consists of one more vehicle if the average demand of product p is higher than is theoretically expected. In section 4.1, the devision of the demand for each customer is explained from which we can calculate the expected average demand.

3 Methodology

3.1 The iterated tabu search algorithm

The iterated tabu search (ITS) proposed to solve the MCMVRP closely resembles the ITS of Silvestrin and Ritt (2017) for the MCVRP, which performed very well compared to earlier proposed heuristics.

Algorithm 1 Iterated tabu search.

procedure ITERATEDTABUSEARCH()
  $s \leftarrow$ initialSolution()
  $s \leftarrow$ tabuSearch(s)
  for $i = 1,...,I$ iterations **do**
    $s' \leftarrow$ perturb(s)
    $s \leftarrow$ tabuSearch(s', i)
    if $F(s) < F(s^*)$ **then**
      $s^* \leftarrow s$
    with probability $(i/I)^2$: $s \leftarrow s^*$
  return s^*

The details of the ITS are shown in Algorithm 1. After constructing an initial solution, it applies the tabu search which will be explained in section 3.1.1. In each iteration of the ITS, the current solution is perturbed, and the tabu search is performed on this perturbation. The next iteration continues with the current found solution s with probability $1 - (i/I)^2$. Otherwise it continues the search from the current best solution s^*. The acceptance criterion was chosen such that the search is diversified at the start while it is intensified around the best found solution at the end.

Initial solution

The initial solution is found by a variation of the sweep algorithm of Gillett and Miller (1974). The customers are ordered in non-decreasing order by their angle in polar coordinates. Then, in this order, the demands of the customers are inserted into the current route until one of the capacity constraints is violated. If a capacity constraint is violated, we insert this customer in a new route. We continue this progress until all customers are assigned to a route. After this clustering, the routes need not to be in an optimal order. Therefore, the order of each route is decided by the nearest neighbor algorithm. We start this algorithm by taking all customers out of the route except the first two customers. The current route then consist of 3 locations: the depot and the two first customers. Then we insert back the next customer in the route, between two locations in the current route, for which the route distance is increased the least. We continue until all customers are again on the route.

Perturbing a solution

At the beginning of each iteration in the ITS, the current solution is perturbed. A random client is selected on a randomly chosen route, and removed from it together with its π neighbors, where π is also randomly chosen in $[0, \lceil\sqrt{n}\rceil]$. If a client is to be removed from the current solution, it is removed from all the routes it is on. The removed clients are in a random order inserted back into the solution in the route which minimizes the insertion cost, where we check all possible insertions between two locations in every route.

3.1.1 The tabu search algorithm

The tabu search allows non-improving moves and avoids cycling between solutions to escape a current local minimum. The algorithm used, again from Silvestrin and Ritt (2017), is detailed in Algorithm 2.

Algorithm 2 Tabu search.

```
procedure TabuSearch(s, i)
    α ← 1
    Choose γ ∈ [0, 1] randomly
    while the incumbent improved in the last √(I − i) iterations do
        s ← bestShiftMove(s)
        Every n_r iterations: s ← refinement(s)
        updatePenalties(α, γ)
        updateTabuList()
    return s*
```

The tabu search starts from a current solution, and makes the best non-tabu move, e.g. the move which reduces the objective value the most. The current solution may exceed the capacity constraints, where the capacity excess is defined as $C^{+}(s) = \sum_{k \in F} \max\{\max_{p \in P} \Delta c_p^k, 0\}$, where $\Delta c_p^k = c(R_k)_p - C_p^k$. The objective value of solution s is then

$$F(s) = d(s) + \alpha C^{+}(s)$$

where α is the penalty for each unit of capacity excess. Initially, we set $\alpha = 1$, and it is updated after a move has been made. If the current solution exceeds the capacity constraints, α is raised by a factor $1+\gamma$. Otherwise, α is decreased by the same factor. The value of γ is at the start of each tabu search randomly chosen in [0,1]. Every n_r iterations, a route refinement is applied to each route, which removes and reinserts the visit which leads to the largest reduction of the route length. The search continues until the incumbent s^* has not been improved the last $\sqrt{I-i}$ iterations, in which case it returns s^*. This stopping criterion has been chosen such that the ITS is given relatively more time in the beginning of the search to improve the incumbent compared to the end of the search.

Making the best move

This section explains the main part of the tabu search: making the best move to the current solution. A move is defined as moving a non-empty subset of demands of a visit to another route. Formally, a move $M = (r, d, v, q)$, where r is the source route, d is the destination route, $v \in r$ is a visit in the source route, and $q \subseteq P(v)$ is a non-empty subset of the demands attended by visit v. If the subset q contains all demands of visit v, the visit is removed entirely from the source route r. Then simply the two neighbors of v are connected to each other. Otherwise, the demands of $P(v) \setminus q$ remain in v in the source route, which does not change the order of the route. The remaining demands q are inserted into the destination route d. Again, if client $V(v)$ is already a part of the destination route d, the demands q are simply added to the existing visit. Otherwise, a new visit $(v(V), q)$ has to be created into the destination route, where we make the least cost insertion between two locations. To reduce to number of available moves, a list of nearest routes is maintained for each client. The nearest route to a client i can be found by first selecting a client j which is closest to client i. Then the route of client j is added to the list only if client j is not on the same route as client i and it is not on the list already. This process is continued until we have found 3 closest routes for each client. The demands of client i are only possibly moved to the routes on this list.

After evaluating every possible move, the move is executed which reduces the objective value $F(s)$ the most. If a move is applied to the current solution, all new moves which involve visit v in source route r are declared tabu for the next τ iterations. At the beginning of each tabu search τ is randomly chosen in $[1, \sqrt{n\bar{r}}]$, where $\bar{r}$ is the number of routes in the initial solution. Declaring moves tabu makes sure cycling between solutions is prevented. If there is no improvement possible, no move will be made. After the penalties are updated, the search for an improving move restarts.

4 Results

4.1 Instances description

The instances we use for the computational experiments are proposed by Christofides, Mingozzi and Toth (1979). While these CMT instances vary in size from 50 to 200 customers, we only use the instances with a maximum of 120 customers for our experiments. The instances are displayed in Table 1, as well as the the shortest known total route length (the best known value, BKV) for the MCVRP found by the ITS of Silvestrin and Ritt (2017).

Table 1: Instances used for computational experiments

CMT	***# Customers***	***BKV***
1	50	546
2	75	863
3	100	833
6	50	558
7	75	946
8	100	888
11	120	1111
12	100	905
13	120	1544
14	100	934

Since the CMT instances are originally created for the VRP, we need to divide the demand c_i for each customer i such that he has a demand for the two available products. We calculate for each customer i its demand for the first product as $c_{i1} = c_i/k$ where $k \in \{3, 4, 5\}$ randomly chosen for each customer. Then the demand for the second product is simply defined as $c_{i2} = c_i - c_{i1}$.

Through the division of the demand for each customer, we calculate the expected average demand for the first product is slightly above 26% of total demand. As mentioned in section 2.2, the average demand is the base in the determination of the capacity devision of the vehicles. As an example, we expect the vehicles to have capacity division as shown in Table 2. Note that this devision differs slightly each time, since the demands are divided randomly for each instance.

Table 2: Example of vehicle compartment division for $\beta = 0.03$

	Average Demand	*Vehicle Type 1*	*Vehicle Type 2*
Product 1	26%	29%	23%
Product 2	74%	71%	77%

4.2 Experiment results

We first present the results of calculating the initial solution for the MCVRP using a variation of the sweep algorithm of Gillett and Miller (1974), as mentioned in section 3. The initial solution given in Table 3 is the average of running the instance 100 times, since each time the instance ran with different random seeds. It is seen that the initial solution is improved by the ITS significantly for all instances. Figure 1 shows the improvement made in each iteration of the ITS on instance CMT-1, where iteration 0 equals the initial solution. The largest improvement, about 15%, is made in the first iteration of the ITS. We get the same result for all other CMT instances: on average, about 85% of the total improvement made by the ITS is made in the first iteration. Looking at Figure 1, we do not expect the ITS to improve the found solution significantly if we increase the number of iterations.

Table 3: Improvement made by applying ITS on the initial solution for the MCVRP

CMT	***Initial Solution***	***ITS***	***% Improved***
1	681	556	18,36
2	1009	886	12,19
3	1006	884	12,13
6	681	571	16,15
7	1008	906	10,12
8	1011	890	11,97
11	1491	1328	10,93
12	1089	961	11,75
13	1516	1310	13,59
14	1095	1002	8,49

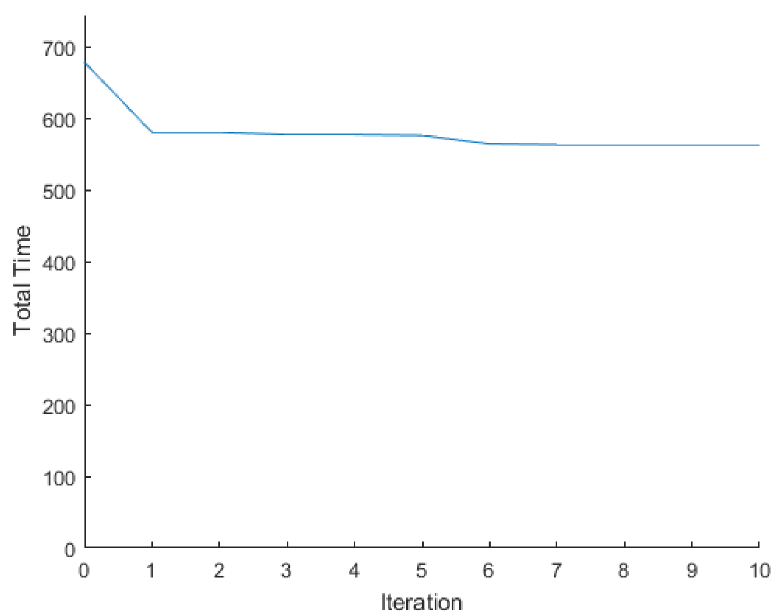

Figure 1: Improvement made by applying ITS on the initial solution per iteration for CMT-1

In Table 4 the value of ITS_β represents the best found solution of our ITS with $\beta \in \{0, 0.01, 0.03, 0.05\}$. For each instance, we ran ITS_β 3 times for 10 iterations. Note that ITS_0 is the same as the ITS in Table 3, since for $\beta = 0$, the MCMVRP reduces to the MCVRP. The computation time of the instances is also given, where the computation time is roughly the same if we run the ITS with different values of β.

Table 5 states the relative deviation of our ITS for the MCVRP to the BKV of Silvestrin and Ritt (2017), where a negative number shows an improvement of the best known value. Table 6 shows the relative deviation of the ITS on the MCMVRP, with $\beta \in \{0.01, 0.03, 0.05\}$ to the ITS on the MCVRP. Again, a negative number indicates the solution has been improved by using a non-homogeneous fleet instead of using identical vehicles as in the MCVRP.

Table 4: Results of ITS on instances for MCVRP and MCMVRP

CMT	*BKV*	ITS_0	$ITS_{0.01}$	$ITS_{0.03}$	$ITS_{0.05}$	*t(min)*
1	546	556	558	578	585	1
2	863	886	893	940	1009	4
3	833	884	888	915	918	8
6	558	571	563	564	576	1
7	946	906	878	916	960	6
8	888	890	887	904	915	8
11	1111	1328	1352	1362	1362	9
12	905	961	1002	1058	1063	6
13	1544	1310	1336	1364	1448	9
14	934	1002	1032	1046	1064	5

Table 5: Comparison of ITS to BKV

CMT	*BKV*	ITS_0
1	546	1,83
2	863	2,67
3	833	6,12
6	558	2,33
7	946	-4,23
8	888	0,23
11	1111	19,53
12	905	6,19
13	1544	-15,16
14	934	7,28

Table 6: Performance non-homogeneous fleet

CMT	ITS_0	$ITS_{0.01}$	$ITS_{0.03}$	$ITS_{0.05}$
1	556	0,36	3,96	5,22
2	886	0,79	6,09	13,88
3	884	0,45	3,51	3,85
6	571	-1,40	-1,23	0,88
7	906	-3,09	1,10	5,96
8	890	-0,34	1,57	2,81
11	1328	1,81	2,56	2,56
12	961	4,27	10,09	10,61
13	1310	1,98	4,12	10,53
14	1002	2,99	4,39	6,19

From Table 5 it is found that the best solution of our ITS deviates on average about 3% to that of Silvestrin and Ritt (2017), where it is even improved with about 4% and 15% for instances CMT-7 and CMT-13 respectively. Although a slight improvement is possible such as for CMT-7, it is highly unlikely that it would improve with as much as 15% for CMT-13. It has been validated however that this solution does not exceed any of the constraints. Since the other instances give reliable solutions using the same ITS, it is more likely that something is wrong with the dataset of CMT-13. This remains to be confirmed, since there was no time to thoroughly investigate the instance data.

From Table 6, we can see that the use of a non-homogeneous fleet managed to improve the solution for some of the instances, with the largest improvement of about 3% for CMT-7 with $\beta = 0.01$. However, using a non-homogeneous fleet with $\beta = 0.01$, increases the solution on average with about 1 %. If we use $\beta = 0.03$ or $\beta = 0.05$, the solution is even worsened by respectively about 4% and 6%. Also, we observe a general pattern within the use of a non-homogeneous fleet for each instance: if we increase β, the best solution found by the ITS is also increased. This suggests that therefore the deviation of the capacity of the vehicles from the average demand should be minimized to get the best solutions by our ITS.

5 Conclusions

The ITS proposed in this research found on average less good solutions for the MCVRP due to a few simplifications opposed to the ITS of Silvestrin and Ritt (2017). Furthermore, although the ITS ran for only 10 iterations, it is expected that an increase in the number of iterations will not improve the best found solution significantly, since on average about 85 % of the total improvement in the first 10 iterations is made in the first iteration.

This research mainly aimed to improve the solution of the MCVRP by changing the fleet of vehicles. The current composition of the non-homogeneous fleet of two different vehicle types could not improve this solution however for several different vehicle compartment devisions. We conclude that the compartment devision of the fleet vehicles which resembles the average demand give better solutions for the used CMT instances.

Further research could be done on the impact of using a non-homogeneous fleet by changing the number of different vehicles. Possibly the effect of using a non-homogeneous fleet pays off when handling a large number of different products.

References

Christofides, N., Mingozzi, A. & Toth, P. (1979). The vehicle routing problem. *Combinatorial Optimization.* Chichester: Wiley, 315-338.

Gillett, B.E. & Miller, L.R. (1974). A heuristic algorithm for the vehicle-dispatch problem. *Operations Research*, 22(2), 340–349.

Sivestrin, P.V. & Ritt, M. (2017). An iterated tabu search for the multi-compartment vehicle routing problem. *Computers and Operations Research*, 81, 192-202.