Vehicle Routing Problem
with Synchronization at Variable Points
Technical Report 2015/1

Dorota Slawa Mankowska, Christian Bierwirth
School of Economics and Business,
Martin-Luther University Halle-Wittenberg, Germany,
dorota.mankowska@wiwi.uni-halle.de,
christian.bierwirth@wiwi.uni-halle.de

Abstract. This article addresses Vehicle Routing Problem with synchronization option to enable transferring cargo among the vehicles at variable points. These points, at which cargo transfer takes place, are an outcome of the planning process. Thus, several further decisions have to be made in addition to the standard VRP decision. It means that not only a decision on which vehicle moves to which customer and when do the vehicles arrive at the customers but also, for example, which synchronization point is suitable for cargo transfer or which vehicle supply another vehicle. Due to the optional cargo transfer, further questions must be answered. On one hand, the synchronization can be beneficial from the cost point of view, e.g. if vehicles due to the cargo transfer option effect less cost. On the other hand, synchronization may be necessary, if due to e.g. deficient infrastructure or insufficient current load, one of the vehicles cannot visit the customer. This paper presents a mathematical formulation for a Vehicle Routing Problem with cargo transfer driven synchronization. Also a heuristic solution method is presented together with first numerical results.

1 Introduction

One of the currently established extensions of Vehicle Routing Problems is the synchronization of vehicle routes. Synchronization occurs if two or more vehicles visit a location either at the same time or in a given sequence. It means that the routes depend on each other according to their spatial and temporal dimension and, thus, they must be planned together. Synchronization is a common concept in the field of transport management. Here, vehicles meet, for example, to exchange cargo among them or to transfer cargo from one vehicle to another vehicle. An interesting real world example is found in waste collection service, see [Del Pia und Filippi, 2006]. Here, two types of vehicles are available: a large and a small vehicle. The substantial size of the large vehicle makes it impossible to traverse the narrowest streets in the city center. To the contrary, the small vehicle can easily drive into any street but due to the size, it attains its capacity limit very quickly. Hence, synchronization between the two vehicles is
to be determined by the planners and incorporated within the vehicles’ routes to enable the small vehicle to dump its collected waste into the large vehicle. The advantage of synchronization in this case is that both vehicles can continue collecting waste without moving back to the landfill site located outside of the city.

Another example can be found in so-called ”truck-meets-truck” traffic, where trucks swap their cargo at a synchronization point and deliver the partner’s shipment to its destination. Synchronization can be beneficial if one truck driver has to deliver goods placed in a swap trailer from destination A to destination B and the other truck driver one from B to A. In such a case, it is reasonable to arrange the synchronization, for example at a half way point, to enable them exchanging the swap trailers and go back to their home places. Except the obvious advantage for the truck drivers, the logistics service provider benefits from the synchronization due to a more efficient use of resources, e. g. reduction of the number of unproductive empty trips. In synchronization at variable points, two (or more) vehicles can meet at the same time at a location which is an outcome of the planning. It means that the location for, for example, transferring cargo from one vehicle to the other, is not known before the planning.

These examples show, that synchronization can be beneficial on the one hand for the overall routing process and on the other hand, synchronization may be necessary due to deficient infrastructure or unfortunate constellation of time windows. If at least one of such situations occur, following decision have to be made in addition to the standard VRP decisions. These are:

i. which vehicles have to meet?
ii. when do these vehicles have to meet?
iii. where is the synchronization point?
iv. which role do which vehicle play (receiving or supplying)?
v. how many cargo units should be transfer from one vehicle to another?

This paper is structured as follows. In section 2, literature on synchronization at variable points is briefly discussed. In section 3, mathematical programming model is presented. For solving this problem, a metaheuristic solution approach is developed in section 4 together with the neighborhood structures and local search procedure. The results of numerical experiments are summarized in section 5. This paper is concluded in section 6.

2 Literature Review

Scientific literature on vehicle routing problems with synchronization at variable points distinguishes two cases:

1. Vehicles exchange the cargo or the cargo cannot remain at the synchronization point, i. e. the vehicles have to meet simultaneously. This case occurs if vehicles cannot load and unload their cargo themselves and no crane is available at a synchronization point permanently.
2. If vehicles can load and unload their cargo themselves (either with help of an available crane or due to usage of easy to exchange swap trailers) and, additionally, it is ensured that the cargo left for the receiving vehicle is secured at a synchronization point, cargo transfer does not have to occur simultaneously. The only requirement here is that the receiving vehicle can take up the cargo after a supplying vehicle has dropped the cargo off. Here, the temporal precedence relation is determined by the role that is assigned to the vehicles, i.e. whether it provides or receives cargo.

For the "truck-meets-truck" traffic, where vehicles usually meet simultaneously at a half-way point of two spatially opposing routes and exchange the load (or the drivers), [Weise et al., 2009] present a multi-depot VRP with synchronization constraints. This concept is usually used in the freight transportation planning, where swap trailers are used as a common load unit containing the orders of the customers. The planning problem comprises finding routes for the trucks carrying swap trailers with the orders, where exchange of cargo at so-called "truck-meets-truck" points is allowed. The goal is to minimize the number of undelivered orders, the total distance traveled and the spare capacities. Instances with up to 3000 orders are solved with an evolutionary algorithm. [Balsliemke, 2004] presents a network-flow model for integrated strategic planning for distribution of new furniture and redistribution of old furniture in a hub & spoke distribution network. The goal of the planning is to minimize the overall transportation cost and transshipment cost at the depots and hubs. But, the "truck-meets-truck" is not explicitly incorporated within the model. Nevertheless, he calls attention not only to choosing the right meeting point for the vehicles but also to choosing the right loading vessel and to plan enough time for exchanging the load. [Kunze et al., 2012] present a project, in which a vendor-independent decision support tool for realization dynamic "truck-meets-truck" traffics using telematics and vehicles’ disposition systems is introduced. It is dedicated to small and medium-sized companies which are willing to cooperate. Then, they submit delivery requests to a pool, which is available online for the cooperation partners and the drivers. The presented tool checks for requests that can be synchronized by investigating the start and end points of the deliveries and the corresponding time windows. Afterwards, a synchronization point is recommended so that the vehicles’ detour length is approximately the same. Unfortunately, no implementation details regarding the VRP are provided, because this paper investigates the technical factors and, thus, aims at verifying the practicability of the theoretical proposition.

Another paper dealing with simultaneous synchronization at variable points is presented by [Del Pia und Filippi, 2006] as has been already mentioned in the introduction. The authors model the waste collection problem as a capacitated arc routing problem with mobile depots. In this paper, no mathematical formulation for this problem is presented. A heuristic solution method based on the Variable Neighborhood Descent is proposed to solve this real world problem.

---

1 In the standard multi-depot VRP, vehicles are located at different depots, e.g. the drivers’ home places, see e.g. [Renaud et al., 1996, Cordeau et al., 1997]
under pursuing the objective of minimizing the total travel time. The instances solved with this approach consist of four vehicles and 422 edges in the corresponding network.

In [Mankowska et al., 2012a], a MILP model for solving a VRP with simultaneous synchronization at variable points is presented. The model comprises the case of two heterogeneous vehicles. The objectives are minimizing the total distance traveled by the vehicles, minimizing the maximal tour duration and minimizing the total vehicle waiting times due to time windows and synchronization. The latter goal aims at minimizing the vehicle idle time. The presented model is tested on randomly generated instances with up to 14 customers and 4 potential synchronization points.

To the knowledge of the author, synchronization with precedence relation at variable points has been considered only rarely. [Mankowska et al., 2012b] propose a MILP for two heterogeneous vehicles for the problem. The motivation is a real world problem of setting up DHL parcel stations in urban areas given a fleet of two heterogeneous vehicles. A small vehicle with a capacity of two parcel stations can access all set-up sites in urban areas. This vehicle is replenished by a large vehicle with capacity of twelve parcel stations, which can access only some of the set-up sites. For transferring the parcel stations, the large vehicle can leave them at synchronization points. These stations are taken up on demand by the small vehicle and brought to the set-up sites. DHL’s fleet consists of only two vehicles, because the setting-up process occurs rarely and also it is not a time critical task and, therefore, investments in additional vehicles would not be economically beneficial for DHL. The goal of the problem is to minimize the total distance traveled by the vehicles and minimize the maximal tour duration. The MILP solves instances with up to 2 vehicles, 9 customers, and 2 synchronization points.

3 Mixed integer programming model

The following notation is used to model the routing of vehicles under synchronization at variable points. The set of nodes is $C^0 = \{0\} \cup C \cup S$, where node 0 refers to a depot, $C$ is a set of customers to be served, and $S$ is a set of potential synchronization points, where vehicles can meet to transfer load. For each potential synchronization point, it is specified, if the cargo transfer must occur simultaneously or if cargo can remain for some time. The simultaneous case is further determined by setting the minimal and the maximal time gap between the start times of unloading and loading operations to $\delta^\min_i = \delta^\max_i = 0$, which means that the start time of unloading of a supplying vehicle is the start time of loading of a receiving vehicle. In the second case, the precedence relation is specified "unloading first, loading second", i.e. the supplying vehicle must first unload the cargo and then the receiving vehicle can load the cargo. It is determined by setting the minimal time gap $\delta^\min_i > 0$ as the time needed by the supplying vehicle for unloading operation and by setting $\delta^\max_i > \delta^\min_i > 0$ as the maximal time within which the cargo can be stored at the transferring point.
Each customer \(i \in \mathcal{C}\) demands \(q_i\) units of a good that is distributed by the vehicles. The delivery must take place within a service time window \([e_i, l_i]\). At the depot, a heterogeneous fleet \(V\) of vehicles is available. To each vehicle \(v \in V\), corresponding costs per kilometer are expressed by \(c_v\). The capacity of each vehicle is \(\text{Cap}_v\). Of course, \(q_i \leq \max_{v \in V} \{\text{Cap}_v\}\) for all \(i \in \mathcal{C}\). Furthermore, we denote by \(p_{iv}\) the service duration that is needed, if customer \(i \in \mathcal{C}\) is served by vehicle \(v \in V\).

To model the accessibility of customers \(i\) by vehicles \(v\), we introduce a binary \(a_{iv}\) with \(a_{iv} = 1\), if vehicle \(v\) can move to customer \(i\). If vehicle \(v\) cannot access customer \(i\), e.g. due to deficient infrastructure, then \(a_{iv} = 0\). Clearly, for the depot and the potential synchronization points in \(S\) we have \(a_{iv} = 1\), for all \(i \in \{0\} \cup S\) and all \(v \in V\). The distances between nodes \(i, j \in \mathcal{C}_0\) are denoted \(d_{ij}\). We assume that travel time units are equal to the distance units.

The following decision variables are used. Binary variable \(x_{ijv}\) is set to 1 if vehicle \(v \in V\) moves directly from node \(i \in \mathcal{C}_0\) to node \(j \in \mathcal{C}_0\), 0 otherwise. The time variable \(t_{iv}\) denotes the operation start of vehicle \(v\), if \(i\) refers to a customer, the start of a load exchange operation, if \(i\) refers to a synchronization point, and the return time, if \(i\) corresponds to the depot. The binary variables \(u_i\) take value 1, if the vehicles use the potential synchronization point \(i \in S\) for transferring load on their routes. \(Q_{iv}\) denotes the amount of cargo of vehicle \(v\) upon arrival at node \(i \in \mathcal{C}_0\). Binary variables \(\alpha_{iv}\) determine whether or not a vehicle \(v\) is at the transferring point a supplying (\(\alpha_{iv} = 0\)) or receiving vehicle (\(\alpha_{iv} = 1\)). We consider the following three objectives for the routing problem:

- Minimizing the total travel costs by all vehicles (\(Z_1\)), as is of relevance for companies aiming at minimization of travel cost.

\[
\min \rightarrow Z_1 = \sum_{v \in V} \sum_{i \in C_0} \sum_{j \in C_0} c_v \cdot d_{ij} \cdot x_{ijv} \quad (1)
\]

- Minimizing the maximal tour duration \(T\) (\(Z_2\)), as is of relevance for companies aiming at even distribution of vehicles’ working hours.

\[
\min \rightarrow Z_2 = T = \max_{v \in V} \{t_{0v}\} \quad (2)
\]

- Minimizing total vehicle waiting times due to time windows and synchronization (\(Z_3\)), as is of relevance for companies aiming at minimizing vehicle idle time.

\[
\min \rightarrow Z_3 = \sum_{v \in V} W_v \quad (3)
\]

The problem is considered under following constraints. Constraints (4) ensure that each vehicle starts and ends its tour at the depot.

\[
\sum_{j \in C_0} x_{0jv} = \sum_{j \in N} x_{j0v} = 1 \quad \forall v \in V \quad (4)
\]

Constraints (5) ensure that every customer is visited exactly once by one of the vehicles.

\[
\sum_{v \in V} \sum_{j \in C_0} x_{ijv} = 1 \quad \forall i \in \mathcal{C} \quad (5)
\]
Constraints (6) balance the flow by ensuring that every vehicle $v$ leaves a node if it visits it.

$$\sum_{j \in \mathcal{C}^0} x_{jiv} = \sum_{j \in \mathcal{C}^0} x_{ijv} \quad \forall i \in \mathcal{C}^0, \ v \in \mathcal{V} \quad (6)$$

Constraints (7) determine the start times of the operations at each node. For tight model formulation $M_1$ is set to $l_i$ if $i \in \mathcal{C}$ or to the end of planning horizon $M_1 = T_{\text{max}}$ if $i \in \mathcal{S}$. This constraints avoid also cycles in the routing.

$$t_{iv} + (p_{iv} + d_{ij}) \cdot x_{ijv} \leq t_{jv} + M_1 \cdot (1 - x_{ijv}) \quad \forall i, j \in \mathcal{C}^0, v \in \mathcal{V} \quad (7)$$

Constraints (8) ensure that the delivery operations start within the time windows of the customers.

$$e_i \leq t_{iv} \leq l_i \quad \forall i \in \mathcal{C} : \sum_{j \in \mathcal{C}^0} x_{ijv} = 1, v \in \mathcal{V} \quad (8)$$

Constraints (9) model the change of the load of vehicle $v$ resulting from serving customers. Here, $M_2$ is set to $\text{Cap}_v$.

$$Q_{iv} - q_i \geq Q_{jv} - M_2 \cdot (1 - x_{ijv}) \quad \forall i \in \mathcal{C}, j \in \mathcal{C}^0, v \in \mathcal{V} \quad (9)$$

Constraints (10) ensure that always two vehicles move to a transferring point if this synchronization point is selected within the planning.

$$\sum_{v \in \mathcal{V}} \sum_{j \in \mathcal{C}} x_{jiv} = 2 \cdot u_i \quad \forall i \in \mathcal{S} \quad (10)$$

Constraints (11) determine for every vehicle $v \in \mathcal{V}$, if it is a supplying or receiving vehicle at a transferring point $i \in \mathcal{S}$.

$$\sum_{j \in \mathcal{C}^0 \setminus \mathcal{S}} q_j \cdot x_{ijv} \leq Q_{iv} + M_2 \cdot \alpha_{iv} \quad \forall i \in \mathcal{S}, v \in \mathcal{V} \quad (11)$$

Constraints (12) ensure that only one vehicle at a transferring point $i \in \mathcal{S}$ is the supplying vehicle and also that vehicles meet, if and only if cargo between them takes place.

$$\sum_{v \in \mathcal{V}} \alpha_{iv} = u_i \quad \forall i \in \mathcal{S} \quad (12)$$

Constraints (13) ensure that the role of the vehicles (supplying or receiving) is only assigned if the vehicles move to the synchronization point.

$$\alpha_{iv} \leq \sum_{j \in \mathcal{C}^0} x_{jiv} \quad \forall i \in \mathcal{S}, v \in \mathcal{V} \quad (13)$$

Constraints (14) and (15) model the time dependency of the unloading and loading operations at synchronization point $i \in \mathcal{S}$ based on the role of the vehicle.
α_{iv2}, on whether or not the transferring point is used in the solution \( u_i \) and on the two vehicles involved in the synchronization.

\[
t_{iv2} - t_{iv1} \geq \delta_i^{\min} - M_3 \cdot (4 - \alpha_{iv2} - u_i - \sum_{j \in C^0} x_{ijv1} - \sum_{k \in C^0} x_{ikv2}) \quad \forall i \in S, v_1, v_2 \in V : v_1 \neq v_2
\]

(14)

\[
t_{iv2} - t_{iv1} \leq \delta_i^{\max} + M_4 \cdot (4 - \alpha_{iv2} - u_i - \sum_{j \in C^0} x_{ijv1} - \sum_{k \in C^0} x_{ikv2}) \quad \forall i \in S, v_1, v_2 \in V : v_1 \neq v_2
\]

(15)

Constraints (16) balance the cargo transfer process at a synchronization point \( i \in S \).

\[
Q_{iv1} + Q_{iv2} \geq Q_{jv1} + Q_{kv2} - M_5 \cdot (3 - u_i - x_{ijv1} - x_{ikv2}) \quad \forall i \in S, j, k \in C^0, v_1, v_2 \in V : v_1 \neq v_2
\]

(16)

Constraints (17) determine the value of the latest arrival time at the depot.

\[
T \geq t_{0v} \quad \forall v \in V
\]

(17)

Constraints (18) determine the value of the total waiting times for vehicle \( v \in V \).

\[
W_v = t_{0v} - \sum_{i \in C^0} \sum_{j \in C^0} (p_{iv} + d_{ij}) \cdot x_{ijv} \quad \forall v \in V
\]

(18)

The domains of the decision variables are modeled in (19)-(22). Constraint (19) avoid traveling to customers, which are inaccessible for vehicle \( v \in V \).

\[
x_{ijv} \in \{0, a_{iv} \cdot a_{jv}\} \quad \forall i, j \in C^0, v \in V
\]

(19)

\[
u_i, \alpha_{iv} \in \{0, 1\} \quad \forall i \in S, v \in V
\]

(20)

\[
t_{iv} \geq 0 \quad \forall i \in C^0, v \in V
\]

(21)

\[
Cap_v \geq Q_{iv} \geq 0 \quad \forall i \in C^0, v \in V
\]

(22)

4 Solution Method

In the following, a heuristic solution method for the presented problem is proposed together with an initial solution procedure (4.1) and neighborhood structure structures for the local search procedure (4.2)
4.1 Initial Solution

The idea of the initial solution algorithm is to assign the customers, which are sorted by urgency, to vehicles, such that the vehicles with the lowest transportation cost per distance unit are preferred in the solution.

The initial routing can be constructed by means of the following heuristic. First, the customers are sorted by increasing end of their time window. This gives priority to urgent customers in the subsequent iterative construction process. Next, the algorithm sorts the vehicles from the smallest to the largest cost rate. This is advantageous in terms of a low possible cost. Next, the algorithm assigns each customer to the route of a vehicle depending on the reachability (time and accessibility) and current load of the vehicle. If the currently considered vehicle has enough load to fulfill the demand requirement of the customer, this customer is assigned to the vehicle. If not, then two possibilities for synchronization open up. If, for example, it pays off to add a synchronization point for load transfer option in terms of lowering the objective value, then a synchronization point is added into the routes of two or more vehicles. For this purpose, the following actions have to be conducted:

(a) the so-called "expected demand" (ED) has to be estimated. It is computed by adding the demand of the subsequent customers from the sorted list, which can be reached within the time window up to the capacity limit of the vehicle.

(b) for all other vehicles has to be checked if current load of the currently considered vehicle is compatible with the ED, i. e. the sum of the current load of the considered vehicle and the potential synchronization vehicle $\tilde{v}$, which can supply the considered vehicle, should be greater or equal to the ED ($load_v + load_{\tilde{v}} \geq ED$)

(c) Taking ED into consideration, for all potential synchronization points, the reachability in terms of the given time windows have to be checked, i. e. if the detour following from inserting a synchronization point for load transfer still enables the receiving vehicle to reach the considered customer within the time window

(d) the potential savings cost has to be determined for each vehicle and each potential synchronization point by comparing the direct way of a vehicle $\tilde{v}$,

![Fig. 1. Synchronization of two vehicles.](image-url)
that can supply the considered customer, to the detour cost of synchronizing the two vehicles \((v \text{ and } \tilde{v})\) at a potential synchronization point, see Fig. 1. Here, the cost of direct delivery are equal to \(d_\emptyset = c_\emptyset \cdot d_{jk}\) and the cost of delivery via synchronization point are equal to \(d_s = c_v \cdot d_{is} + c_\tilde{v} \cdot d_{js} + c_v \cdot d_{sk}\) and the performance indicator is equal to \(\Delta_{sv\tilde{v}} = d_\emptyset - d_s\). This has to be done for all potential synchronization points and for all vehicles, that would be able to supply the considered customer within the time window and if the ED can be reached.

If all of the actions has a feasible outcome and it exists at least one negative saving, then a synchronization point - with the lowest negative \(\Delta_{sv\tilde{v}1}\) - is added to the routes of \(v\) and \(\tilde{v}\).

Otherwise, synchronization may be necessary. This situation occurs if

- the demand of the currently considered customer is greater than the maximum of the current load of all vehicles.
- the currently considered customer is the last customer

In such a case, all actions (a)-(d) have to be conducted and a synchronization point with the lowest \(\Delta_{sv\tilde{v}}\) is added to the routes of vehicles \(v\) and \(\tilde{v}\) even if \(\Delta_{sv\tilde{v}}\) is positive. Note, that it can happen that more than one vehicle has to supply the currently considered vehicle, then the vehicles meet together at the synchronization point.

If synchronization is not beneficial and also not necessary and the current vehicle has not enough load to supply the currently considered customer or it will not reach this customer within the time window, the algorithm switches to the next vehicle in the sorted list and this new vehicle becomes the currently considered vehicle. The outline of the heuristic is shown in Fig. 2.

The presented procedure cannot always find a feasible routing because of infeasible configuration of the time windows. In such a case, penalty cost for delays in delivery can be added into the objective function, which gets higher priority within the local search procedure than the actual objective value.

### 4.2 Neighborhood Structures

In the following, we present the neighborhood structures which are used for the local search procedure. The neighborhood structures are defined for the matrix solution representation as presented in [Mankowska et al., 2014]. The moves are based on intra, inter, switch, and swap moves, as is presented in Fig. 3. After each move, the feasibility of the routing has to be investigated. For this purpose, not only the compatibility with the given time windows has to be checked but also the load of the vehicles upon arrival at each customer. If the neighbor solution is not feasible a repair procedure is applied to prove whether or not this solution can be repaired by adding synchronization point into the routing. It adds synchronization points as has been described within the initial solution. If the solution cannot be repaired with this procedure, it will not be used within further local search procedure and such a move is forbidden.
INITIAL ROUTING
1: evaluate \( sort_c \) \( \triangleright \) Sort customers from smallest to largest \( l_i \)
2: evaluate \( sort_v \) \( \triangleright \) Sort vehicles from smallest to largest \( c_v \)
3: set \( load(v) = Cap_v \) \( \triangleright \) Current load of the vehicles is equal to their capacity limit
4: for \( k = 1 \rightarrow |\mathcal{V}| \) do
5: \( v = sort_v(1) \) \( \triangleright \) start with the vehicle with the lower \( c_v \)
6: if \( v \) can reach \( sort_c(k) \) by \( l(sort_c(k)) \) and \( a_{sort_c(k)} \) then
7: if current load of \( v \geq q_{sort_c(k)} \) then
8: \( o_v = sort_c(k) \)
9: update \( load(v) = load(v) - q_{sort_c(k)} \)
end if
10: if \( load(v) < q_{sort_c(k)} \) then
11: find the expected demand (ED) \( \triangleright \) \( \forall v \in \mathcal{V} \) : \( \emptyset \neq v \neq v \)
12: compute the detour cost (\( \Delta_{sv(v)} \)) \( \triangleright \) check for compatibility with time windows
end for
13: \( \forall k = 1 \rightarrow |\mathcal{V}| : \emptyset \neq \emptyset \neq \emptyset \)
14: \( \forall v = 1 \rightarrow |\mathcal{V}| : \emptyset \neq v \neq v \)
15: check for compatibility with time windows
end for
16: \( \forall \emptyset : \emptyset \neq \emptyset \)
17: \( \forall k = 1 \rightarrow |\mathcal{V}| : \emptyset \neq k \neq k \)
18: \( \forall o_k = sort_v(k) \)
19: \( load_{diff} = ED - load(v) \)
20: \( load(v) = load(v) + load_{diff} \)
21: \( load(v) = load(v) + load_{diff} \)
22: \( load(v) = load(v) + load_{diff} \)
end if
23: if \( k = |\mathcal{V}| \) or \( q_{sort_c(k)} \geq Cap_{sort_c(k)} \) then
24: synchronize them at the best synch point \( \triangleright \) \( \forall v \in |\mathcal{V}| \) : \( \emptyset \neq v \neq v \)
25: \( \forall v \in |\mathcal{V}| : \emptyset \neq v \neq v \)
26: \( \forall v \in |\mathcal{V}| : \emptyset \neq v \neq v \)
27: \( \forall v \in |\mathcal{V}| : \emptyset \neq v \neq v \)
28: \( \forall v \in |\mathcal{V}| : \emptyset \neq v \neq v \)
29: \( \forall v \in |\mathcal{V}| : \emptyset \neq v \neq v \)
30: \( \forall v \in |\mathcal{V}| : \emptyset \neq v \neq v \)
end if
31: end if
32: end if
33: if \( v \) cannot reach \( sort_c(k) \) by \( l(sort_c(k)) \) then
34: \( v++ \)
end if
37: if \( v = |\mathcal{V}| \) then
38: \( \forall v = 1 \rightarrow |\mathcal{V}| : \emptyset \neq v \neq v \)
39: compute the detour cost (\( \Delta_{sv(v)} \)) \( \triangleleft \) check for compatibility with time windows
40: end for
43: \( \forall \emptyset : \emptyset \neq \emptyset \)
44: \( \forall o_k = sort_v(k) \)
45: \( load_{diff} = ED - load(v) \)
46: \( load(v) = load(v) + load_{diff} \)
47: \( load(v) = load(v) + load_{diff} \)
48: \( load(v) = load(v) + load_{diff} \)
end if
49: if \( k = |\mathcal{V}| \) or \( q_{sort_c(k)} \geq Cap_{sort_c(k)} \) then
50: synchronize them at the best synch point \( \triangleright \) \( \forall v \in |\mathcal{V}| \) : \( \emptyset \neq v \neq v \)
51: \( \forall v \in |\mathcal{V}| : \emptyset \neq v \neq v \)
52: \( \forall v \in |\mathcal{V}| : \emptyset \neq v \neq v \)
53: \( \forall v \in |\mathcal{V}| : \emptyset \neq v \neq v \)
54: \( \forall v \in |\mathcal{V}| : \emptyset \neq v \neq v \)
55: \( \forall v \in |\mathcal{V}| : \emptyset \neq v \neq v \)
end if
end if
end if
end for
59: end for
60: return \( \mathcal{O} \)

Fig. 2. Algorithm for constructing an initial solution.
5 Numerical Experiments

In the following, numerical experiments are presented for the generated test instances as follows. Two sets containing 10 test instances each are considered. In set A, the number of customers is set to 10, there is one synchronization point which can be used multiple number of times, 3 vehicles of capacity 10, 15 and 20 respectively. In set B, the number of customers is 25, there are 5 vehicles with capacities 10, 10, 15, 15, 20. For both sets, the demand of the customers ranges from 1 to \( \sum_{v=1}^{|V|} \text{Cap}_v / |C| \). The cost per distance unit corresponding to the use of the vehicle \( c_v \) are randomly generated and drawn from an interval corresponding to the capacity of the considered vehicle and the vehicle with the next smallest capacity, i.e. \( c_1 \) ranges from 1 to \( \text{Cap}_1 \), \( c_2 \) ranges from \( c_1 + 1 \) to \( \text{Cap}_2 \) and so on. The accessibility of the vehicles is not limited, i.e. all vehicles can reach all customers. The customers are randomly placed in an area 100 × 100 and the distances are assumed to be Euclidean. Processing times for unloading the cargo at each customer are randomly drawn from an interval [10, 15]. For synchronization purposes, the minimal \( \delta_{\text{min}} \) and the maximal time gap \( \delta_{\text{max}} \) within which the synchronization has to take place is randomly drawn from an interval [1, 60] and [\( \delta_{\text{min}} \), \( \delta_{\text{min}} + 60 \)], respectively. The time windows are of length 120, where the beginning of each time window is a randomly drawn integer from [0, 480]. The components of the objective function are weighted with one.
We conduct two tests. In the first test, we run Cplex for each instance from set $A$ and $B$ with a computation time limit set to one hour. In the second test, the presented metaheuristic solution method is run for each instance from set $A$ and $B$. The results of both numerical tests are presented in Table 1. For the first test, the upper bound (UB), lower bound (LB), the gap between the upper and lower bound (gap) as well as the computation time in seconds (cpu) are reported. For the second test, the initial solution (init sol), best solution (best sol) and the computation time in seconds (cpu) are presented. To investigate the solution quality delivered by the metaheuristic, the gaps to the upper and lower bound are measured. Also, for each set $A$ and $B$ an average values of the investigated performance measure indicators are listed.

The numerical experiments show that Cplex solver struggles with finding an optimal solution within one hour computation time even for instances with very small amount of customers and vehicles (set $A$). It means that the complexity of the models is colossal. The proposed heuristic method delivers solution of rather poor quality but still outperforms Cplex in the computational time and it finds for some instances solutions of better quality.

<table>
<thead>
<tr>
<th>instance</th>
<th>Cplex</th>
<th>AVNS</th>
<th>gap to</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UB</td>
<td>LB</td>
<td>gap</td>
</tr>
<tr>
<td>$A_1$</td>
<td>6752.6</td>
<td>2900.5</td>
<td>57.0%</td>
</tr>
<tr>
<td>$A_2$</td>
<td>7058.4</td>
<td>4320.8</td>
<td>38.8%</td>
</tr>
<tr>
<td>$A_3$</td>
<td>5465.5</td>
<td>3001.1</td>
<td>45.1%</td>
</tr>
<tr>
<td>$A_4$</td>
<td>5818.9</td>
<td>1683.3</td>
<td>71.1%</td>
</tr>
<tr>
<td>$A_5$</td>
<td>7285.1</td>
<td>3522.0</td>
<td>51.7%</td>
</tr>
<tr>
<td>$A_6$</td>
<td>6009.3</td>
<td>2582.2</td>
<td>57.7%</td>
</tr>
<tr>
<td>$A_7$</td>
<td>6366.6</td>
<td>3993.4</td>
<td>37.3%</td>
</tr>
<tr>
<td>$A_8$</td>
<td>7225.2</td>
<td>4536.1</td>
<td>37.2%</td>
</tr>
<tr>
<td>$A_9$</td>
<td>7700.1</td>
<td>5016.2</td>
<td>34.9%</td>
</tr>
<tr>
<td>$A_{10}$</td>
<td>4622.9</td>
<td>1541.2</td>
<td>66.7%</td>
</tr>
</tbody>
</table>

| $∅$      | 6439.5| 3309.7| 49.7%  | 3600 | 8317.3   | 7174.3   | < 1  | 11.0% | 53.5%|

| $∅$      | -     | -     | -      | -    | -        | -        | -    | -     | -     |

Table 1. Numerical results.
For set $B$, Cplex fails in finding any feasible solution within one hour computation time. Also, the delivered lower bounds are of poor quality. The presented solution method delivers solutions within at most 19 seconds, which makes this method applicable to the real world problems.

To summarize, the presented solution method delivers feasible solutions of rather week performance but still it delivers feasible solutions. In addition, this method can be easily extended by adding new neighborhood structures into the searching process. Also, a new method of adding synchronization points into the solutions should be investigated to better estimate the traveling cost according to the whole routing and not only to the current situation as has been proposed. Also, the procedure of computing expected demand (ED) can be performed in another way, e.g. at most $\alpha$ subsequent customers are taken into consideration, where $\alpha$ is then variable.

6 Conclusion

In this paper, VRP with synchronization at variable points is investigated. For this problem, a mathematical model together with a metaheuristic solution method were proposed. The standard Cplex solver struggles when solving the model even for very small test instances. The developed solution approach delivers feasible solutions but the quality of them is rather on a medium level. Thus, the future research should concentrate on finding more powerful solution method to tackle this problem.

Acknowledgments: This research is funded by the Deutsche Forschungsgemeinschaft (DFG) under project reference B02110263.

References


