Oper Res Int J (2020) 20:2085–2108
https://doi.org/10.1007/s12351-018-0403-4

ORIGINAL PAPER

# Two metaheuristic approaches for solving the multi-compartment vehicle routing problem

**Hiba Yahyaoui[1,2] · Islem Kaabachi[1] · Saoussen Krichen[1] · Abdulkader Dekdouk[2]**

Received: 27 September 2017 / Revised: 22 February 2018 / Accepted: 10 May 2018 /
Published online: 8 June 2018

**Abstract** We address in this paper a multi-compartment vehicle routing problem (MCVRP) that aims to plan the delivery of different products to a set of geographically dispatched customers. The MCVRP is encountered in many industries, our research has been motivated by petrol station replenishment problem. The main objective of the delivery process is to minimize the total driving distance by the used trucks. The problem configuration is described through a prefixed set of trucks with several compartments and a set of customers with demands and prefixed delivery. Given such inputs, the minimization of the total traveled distance is subject to assignment and routing constraints that express the capacity limitations of each truck's compartment in terms of the pathways' restrictions. For the *NP*-hardness of the problem, we propose in this paper two algorithms mainly for large problem instances: an adaptive variable neighborhood search (AVNS) and a Partially Matched Crossover PMX-based Genetic Algorithm to solve this problem with the goal of ensuring a better solution quality. We compare the ability of the proposed AVNS with the exact solution using CPLEX and a set of benchmark problem instances is used to analyze the performance of the both proposed meta-heuristics.

**Keywords** Metaheuristics · Multi-compartment · Vehicle routing problem · Variable neighborhood search · Genetic algorithm

✉ Hiba Yahyaoui
hiba.yahyawi@gmail.com

[1] Institut Supérieur de Gestion, LARODEC laboratory, Université de Tunis, Tunis, Tunisia

[2] College of Arts and Applied Sciences, Dhofar University, Salalah, Sultanate of Oman

# 1 Introduction

The multi-compartment vehicle routing problem (MCVRP) is an extension of the capacitated vehicle routing problem (CVRP), where the MCVRP consists of designing a set of minimal cost routes to serve demands for different types products of a set of customers. The MCVRP arises frequently in petrol distribution systems. Our focus in this paper will be devoted to procedures in the petroleum company's operations. (a) Different types of petroleum products have to be delivered by the company to a set of geographically dispersed stations by a fleet of identical tank-trucks departing from a single depot. (b) Each station orders known quantities of one or more products. (c) However, these products are incompatible, they must be transported together in separate compartments of the same truck. (d) Trucks are partitioned into a constant number of compartments with fixed capacities. Each compartment is reserved for only one product. Moreover, all products ordered by each station must be delivered by only one truck. (e) Note that each truck is equipped with a flow metres. This means that the total stations' demand assigned to any route for each product does not exceed the capacity of the reserved compartment for this product. The company's goal is to determine a set of routes satisfying all the demanded products at minimal routing cost and number of trucks (fleet size).

Further applications of MCVRP in the context of petrol station replenishment were presented in the literature, Avella et al. (2004) developed a heuristic and branch-and-price algorithm based on a set partitioning formulation to solve a real-word petrol replenishment problem with several tank-trucks of different types. Ng et al. (2008) studied two small petrol distribution networks in Hong Kong. They presented a model for simultaneously assigning trips to trucks and stations.

The collection of papers by Cornillier et al. (2008a, b, 2009, 2012) have several presented approaches for different versions of the petrol station replenishment problem. In Cornillier et al. (2008a), an exact algorithm was proposed to solve the petroleum distribution problem. Later, heuristics and the exact method of Cornillier et al. (2008a) was used for the multi-period case of the problem were proposed in Cornillier et al. (2008b) Two heuristics based on the mixed linear programming formulation were presented for the problem with time windows in Cornillier et al. (2009). In Cornillier et al. (2012), a heuristic was developed for the multi-depot version of the problem. Popović et al. (2012) addressed the fuel delivery problem characterized as multi-period multi-product inventory routing problem (IRP) with homogenous tank-trucks and deterministic consumption that varies with each station and each product and to solve this problem The authors proposed a variable neighbourhood search (VNS) heuristic, in which the stochastic VNS heuristic showed better results than the deterministic "compartment transfer" (CT) heuristic. More recently, Coelho and Laporte (2015) proposed classification and mathematical models for the different MCVRPs that arises in the distribution of petroleum products. They developed also a branch-and-cut algorithm that is applicable to all of these problems. In the above papers concern the petrol station replenishment problem, which is the control application of our problem, trucks often are not equipped with a flow metre, that's why each compartment that fully contains one product should be delivered and emptied completely during the visit of one station.

Among other real-life applications of the MCVRP are the collection of milk from farms located on the country side, where trucks with multi-compartment prevent different milk types from mixing (see for example Paredes-Belmar et al. 2016). Oppen and Løkketangen (2008) solved the livestock collection problem where different animal categories are transported from ranches to abattoirs on separate compartments. Chajakis and Guignard (2003) targeted the delivery of groceries problem where refrigerated and non-refrigerated grocery items are stored in separate compartments in the same truck. The memetic algorithm and a tabu search were applied to transportation of animal foods to farms in Fallahi et al. (2008). There have been also various approaches to the waste collection problem in (Reed et al. 2014; Abdulkader et al. 2015; Elbek and Wøhlk 2016). Lahyani et al. (2015) proposed a mathematical formulation and branch-and-cut algorithm to solve the multi-period MCVRP in the olive oil collection process in Tunisia.

To the best of our knowledge, few other contributions have considered the variant defined above of the MCVRP and have solved the problem without taking into account its periodicity aspect (Avella et al. 2004; Coelho and Laporte 2015; Lahyani et al. 2015), stochastic demands (Elbek and Wøhlk 2016), time windows (Cornillier et al. 2009), free assignment of products to compartments (Oppen and Løkketangen 2008; Cornillier et al. 2009; Coelho and Laporte 2015). Both these works fall under the category of inventory-routing problems. The different modeling features, optimization models and solution approaches developed for this variant of the MCVRP have been described below.

Fallahi et al. (2008) applied the MCVRP in which the different animal foods is supplied to the farms separately. They proposed three algorithms a constructed heuristic, a memetic algorithm combined with a path relinking method used as post optimization, and a tabu search to solve this problem. These algorithms have been tested using a set of instances obtained by adapting benchmark instances for the classical VRP. Each algorithm is compared with a version in which the orders of customers cannot be split and the multi-compartment version. The multi-compartment version improved the results on average. By comparing two algorithms, the tabu search generated better solutions than memetic algorithm but it required more running time. In addition, the results of the algorithms found very close to the best-known classical VRP solution.

Muyldermans and Pang (2010) proposed a guided local search for the MCVRP and demonstrate the benefits of co-collection of sorted waste from different locations to central location by multi-compartment vehicles over separate collection by vehicles. They compared the results with Fallahi et al. (2008). They assumed that more than one vehicle can visit the customer to fulfill demands of different products.

Dirac (1953) investigated the vehicle routing problem with compartments (VRPC), which generalizes the MCVRP to flexible compartments, and the allocation of several products to a compartment, subject to constraints on the products and compartments incompatibilities. They proposed a heuristic approaches including constructive methods, local searches, solution modification by destruction and reconstruction, and meta-heuristics. The best metaheuristic from the highest performing was the record-to-record-travel accepting solutions with a relative deviation of up to 3%.

Reed et al. (2014) proposed an ant colony system (ACS) with 2-opt local search improvement for the classical CVRP. They extended their approach to solve the MCVRP. They evaluated their algorithm on five sets of instances. Abdulkader et al. (2015) was inspired from the work of Reed et al. (2014) algorithm extension for the MCVRP. They proposed a hybridized ant colony algorithm (HAC) algorithm combined with local search procedures for the MCVRP. They generated new benchmark instances to test the performance of their algorithm. The numerical experiments indicated on average, that the proposed algorithm improved the solutions of Reed et al. (2014) by about 5% using less computational time.

As we are addressing an *NP*-hard optimization problem, since it is a generalization of the vehicle routing problem (Battarra et al. 2014), we were only able to solve problem instances with a limited size to optimality by application of a mathematical model-based exact solution approach. Therefore, an adaptive variable neighbourhood search (AVNS) and a PMX-based genetic algorithm are designed to solve the large scale instances. To evaluate the performance of the proposed algorithms, its results are compared with the results of exact method, existing meta-heuristic algorithms. This paper also contributed on developing a mathematical formulation of the one compartment VRP and used CPLEX and our AVNS to point out the economic benefits of introducing multiple compartments on the vehicles.

The remainder of this paper is organized as follows: The MCVRP is formally defined and modeled as a linear program in Sect. 2. Section 3 presents a description of the AVNS heuristic. The description of the PMX-based genetic algorithm is given in Sect. 4. Computational results are presented in Sect. 5, followed by the conclusions in Sect. 6.

## 2 The MCVRP: description and problem modelling

The MCVRP can be modeled as an undirected graph $G = (V, E)$, where $V = \{0\} \cup V_c$ is the vertex set that includes the depot and the set of customers $V_c$. The edge set $E = \{\{i,j\}|i,j \in V, i \neq j\}$ expresses the distance between each pair of geographical points: a distance $d_{ij}$ is associated to each edge $\{i,j\} \in E$.

Customers can receive a set of $P$ products using available fleet of $K$ homogeneous trucks with capacities $Q_{max}$. Each truck starts its tour from the depot, visit its designated customers, and then turns back to the depot. Trucks are equipped with multiple compartments with a limited capacity, dedicated to different types of products. All trucks have the same number of compartments. Moreover, the number of compartments is equal to the total number of products. This means that all trucks are equipped with $|P|$ compartments with respective capacities $Q_p$ $(p = 1, \dots, |P|)$. Each customer $i \in V_c$ has a known demand $q_{ip} \leq Q_p$, for each product $p$. Indeed, each customer $i$ is visited exactly once by only one truck. The objective of the MCVRP is to minimize the total travel distance for all routes using a minimum number of trucks from an available fleet of $K$ trucks, while satisfying the demands and compartments capacity constraints of the used truck.

The proposed formulation is based on the following assumptions:

- The fleet is homogeneous.
- Trucks have multiple compartments.
- Each truck starts its tour and the end returns to the depot.
- Each compartment is dedicated to one type of product.
- The number of compartments is equal to the number of products.
- Demands of all customers are constant and known in advance and must be fully met.
- Each customer is received different products exactly once by single truck.

The MCVRP formulation requires the following data entries and decision variables:

| Inputs | | Outputs |
|---|---|---|
| $P$ | Set of products | $x_{ijk} = \begin{cases} 1 & \text{if truck } k\text{travels from customer}i \text{ to } j \\ 0 & \text{elsewhere} \end{cases}$ |
| $K$ | Set of trucks | $z_{jkp} = \begin{cases} 1 & \text{if customer } j\text{recieved product } p \text{ from truck } k \\ 0 & \text{elsewhere} \end{cases}$ |
| $V$ | Set of all graph vertices | $y_{ik} = \begin{cases} 1 & \text{if truck } k \text{ delivers customer } i \\ 0 & \text{elsewhere} \end{cases}$ |
| $V_c$ | Set of customers | $u_{ikp}$: Continuous variables that represent the remaining quantity of product $p$ by truck $k$ after leaving customer $i$ |
| $d_{ij}$ | Distance between $i$ and $j$ | |
| $Q_{max}$ | Capacity of each truck | |
| $q_{ip}$ | Customer $i$'s demand for product $p$ | |
| $Q_p$ | Capacity of truck compartment dedicated to product $p$ | |

The following mathematical model (1)–(12) accounts for the MCVRP is available in Kaabachi et al. (2016)

$$Minz(x) = \sum_{i\in V}\sum_{j\in V, j\neq i}\sum_{k\in K} d_{ij}\, x_{ijk} \tag{1}$$

$$S.t.$$

$$\sum_{k\in K} y_{ik} = 1 \quad i \in V_c \tag{2}$$

$$\sum_{k\in K} y_{0k} \leq |K| \tag{3}$$

$$\sum_{i \in V, i \neq j} x_{ijk} = y_{jk} \quad j \in V_c, k \in K \tag{4}$$

$$\sum_{j \in V, i \neq j} x_{ijk} = y_{ik} \quad i \in V_c, k \in K \tag{5}$$

$$\sum_{i \in V, p \in P} q_{ip} \, y_{ik} \leq Q_{max} \quad k \in K \tag{6}$$

$$u_{ikp} - u_{jkp} + Q_p \, x_{ijk} \leq Q_p - q_{ip} \quad i,j \in V_c, i \neq j, k \in K, p \in P \tag{7}$$

$$q_{ip} \leq u_{ikp} \leq Q_p \quad i \in V_c, k \in K, p \in P \tag{8}$$

$$z_{jkp} \leq \sum_{i \in V} x_{ijk} \quad j \in V_c, k \in K, p \in P \tag{9}$$

$$\sum_{k \in K} z_{jkp} = 1 \quad j \in V_c, p \in P \tag{10}$$

$$\sum_{j \in V_c} z_{jkp} \, q_{jp} \leq Q_p \quad k \in K, p \in P \tag{11}$$

$$\begin{aligned} & x_{ijk}, z_{jkp}, y_{ik} \in \{0,1\} \quad i,j \in V, k \in K \\ & \quad j \in V_c, p \in P \end{aligned} \tag{12}$$

- The objective function (1) minimizes the travel distance for all routes performed by the used trucks.
- Constraints (2) and (3) impose that exactly one truck services each customer and a subset of trucks leave the depot.
- Constraints (4) and (5) force each truck to arrive and leave from customer *i* only if it serves that customer.
- Constraint (6) imposes that the total demand delivered by each truck is less than its capacity.
- Inequalities (7) and (8) are subtour elimination constraints imposing both the capacity and connectivity of the feasible routes.
- Constraint (9) states that each customer's demand for each product must be delivered by only one truck.

- Constraint (10) means that each product ordered by a customer is delivered by exactly one truck.
- Constraints (11) checks the compartment capacity constraint for each truck.
- Constraints (12) state the binary nature of the decision variables.

# 3 An adaptive variable neighborhood search for the MCVRP

## 3.1 The variable neighborhood search

The Variable neighborhood search (VNS) is an effective metaheuristic for combinatorial optimization problems proposed by Mladenović and Hansen (1997) (for surveys of VNS, see Hansen et al. 2008a, b). The VNS uses two or more neighborhoods and systematically changes the neighborhood within a local search algorithm.

It explores the different neighborhood structures in both deterministic and stochastic ways. In fact, it chooses a point from the $k^{th}$ neighborhood stochastically, then, it performs a deterministic local search. A deep study of the literature reveals that the VNS is efficient in solving optimization problems (Brimberg et al. 2017; Wang et al. 2017; Rahimian et al. 2017; Wang et al. 2017). Moreover, several VNS variants are proposed for solving large problem instances for routing problems (Bula et al. 2017; Breunig et al. 2016; Li and Tian 2016). For instance, in the local search step of the VNS we can use more than one neighborhood, such variant called the general VNS (GVNS). Besides to that, the proposed approach, called AVNS, disposes a learning stage to enhance its performance. It can provide

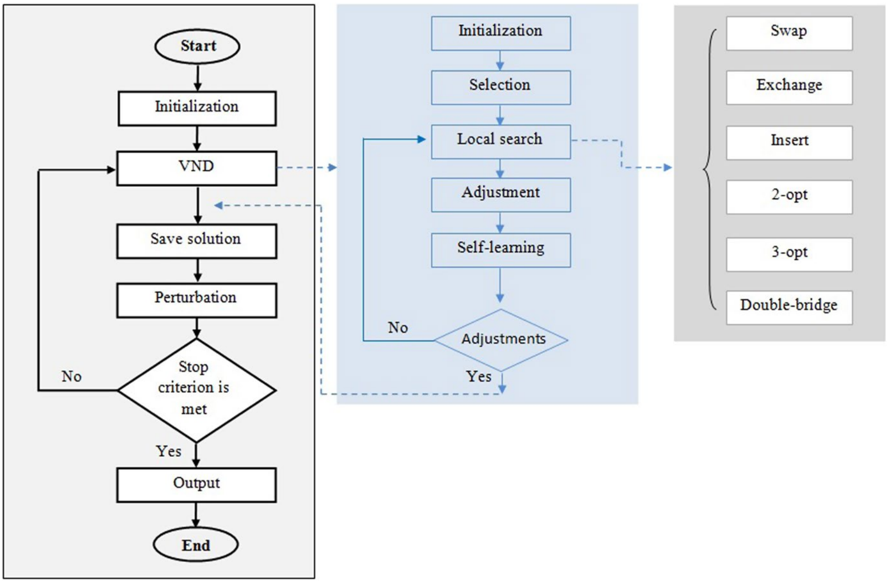


**Fig. 1** The Adaptive variable neighborhood search scheme

a better investigation by obtaining knowledge acquired after interaction with the search space, which lead to a better framework that can generate improved solutions. The main steps of the AVNS are stated in Fig. 1 as follows:

The algorithm starts with a random feasible solution, then it applies a local search procedure that alternates between six neighborhoods. The next step of the AVNS consists on a perturbation operator that aims to diversify the solution. These process is iterates until a termination criterion is met.

### 3.2 Self-learning local search

In the local search procedure we used a variable neighborhood descent with self learning selection applying a set of neighborhoods to explore promising solutions. Furthermore, in order to design an adaptive control mechanism for the neighborhood selection and combining, we assign a weight $w_i^j$ to each one. The weights are designed and updated, so that they reflect the relative performance of each neighborhood during the search, two metrics are proposed:

(1) *improve* is used to enlarge the search space using alternative neighborhood in terms of solution quality. It is denoted $\Delta f_i^j$ and calculated as follows: let an incumbent solution $s^j$ at iteration $j$ and $s'$ be the solution obtained by neighborhood $N_i$, $f$ denotes the fitness value of the solution:

$$\Delta f_i^j = f(s^j) - f(s').$$

(2) *time*: is applied to evaluate the computation time required to improve the current solution called $\Delta t_i^j$. It measures the time difference between the starting operation and finding the obtained solution $s'$ performed by neighborhood $N_i$.

We then calculate the weight ratio as follows: $r_i^j = \frac{\Delta f_i^j}{\Delta t_i^j}$. At each iteration $j$, AVNS

chooses (among neighborhoods where some improvement can still be obtained), the one having the highest weight.

The used neighborhoods are based on five well-known edge-exchange and customer-exchange operators:

*Swap* Two successive customers are swapped, we allowed customers from different vehicles.

*Exchange* Two customers are exchanged. Then the solution is updated regarding the structural constraints.

*Insert* Customer is removed from its current position in a tour and reinserted in the solution.

*2-opt* Two edges are deleted then added in order to obtain lower cost tour, it removes the ”crossings” of edges.

*3-opt* Three edges are deleted then three edges are added to restore the tour. some 3-opt moves are equivalent to apply 2-opt move two times.

*Double-bridge* Four edges are deleted then Four edges are added, but while keeps the orientation of the tour, so it is a special case of the 4-opt neighborhood.

# 4 A PMX-based genetic algorithm for the MCVRP

## 4.1 The genetic algorithm

We develop a partially matched crossover PMX-based GA termed *GAPMX* that operates as follows:

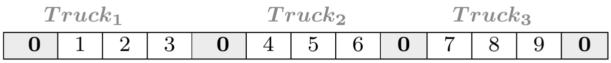
The initial population generation applies the greedy randomized adaptive search procedure (GRASP). Then we apply the genetic operators until a stopping criterion is reached. The *GAPMX* steps are described in Algorithm 1.

**Algorithm 1** *GRASP Genetic algorithm* $(GRASP - GA)$ *for the WSRP*

```
Input:
Generate the initial population $P_0$: $[random\ +\ GRASP]$
while terminate criterion $t$ do
    for $i < p_{size}$ do
        Select $a_i$ as a father and $b_i$ as a mother of $p_i$
        Crossover PMX($x_i$, $x_j$) $\leftarrow a_i$ and $b_i$
        if feasible and improves($x_1$, $x_2$) then
            $x' \leftarrow x$
        end if
    end for
    Mutate $x'$
end while
Output:
```

## 4.2 The GRASP algorithm

The greedy randomized adaptive search procedure (GRASP) is a heuristic algorithm that progressively builds from scratch. It consists of successive constructions of a greedy randomized solution until a complete feasible solution is obtained. The applied greedy randomized procedure introduced in Haddadene et al. (2016) operates as follows: for each available vehicle we assign an initial path. Then, by considering a sorted list of potential customers, these routes are constructed from the list while respecting time windows constraints. For the construction of the candidate list, we used the same procedure as Haddadene et al. (2016). Once all customers are visited, the algorithm stops. This procedure can lead to non feasible solution, that's why additional nodes are inserted. The selection of the next element is determined by the fitness of all candidate solution according to the value of the objective function.



| $Truck_1$ | | | | | $Truck_2$ | | | | $Truck_3$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 2 | 3 | **0** | 4 | 5 | 6 | **0** | 7 | 8 | 9 | **0** |

**Fig. 2** Example of chromosome: permutation encoding

## 4.3 Solution encoding

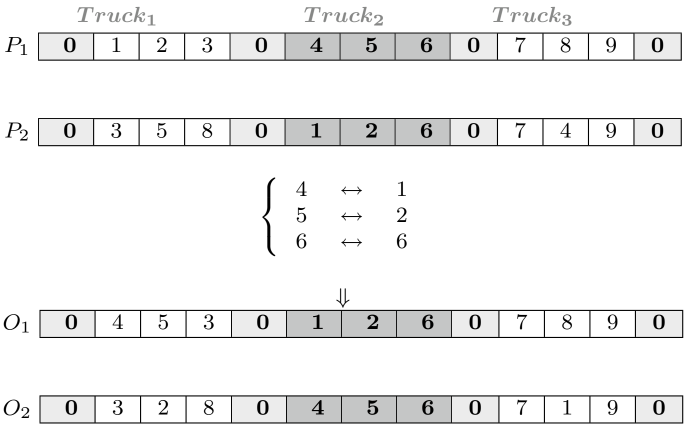
The encoding of a chromosome is reported in Fig. 2 as a $(n + m + 1)$-sized vector that displays the $n$ customers served by $m$ vehicles.

### *4.3.1 Partially matched crossover (PMX)*

Many attempts have been suggested to discover an appropriate crossover operator for the routing problems, hence Goldberg and Lingle (1985) proposed the Partially Matched Crossover (PMX) crossover. In fact, it outperformed other operators in the work of Kumar and Karambir (2012). The mechanism of the PMX crossover, as presented in Fig. 3, builds two offspring as flows. It selects one parent randomly, then it chooses a sub-sequence from the gene strings. A matching selection is provided while preserving the order and position as possible from the other parent.

### *4.3.2 Mutation*

To maintain the genetic diversity from one generation of a population to the next, we use the mutation operator. As reported in Fig. 4, it consists on altering one gene values in a chromosome from its original state.



| | $Truck_1$ | | | | | $Truck_2$ | | | | $Truck_3$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $P_1$ | **0** | 1 | 2 | 3 | **0** | **4** | **5** | **6** | **0** | 7 | 8 | 9 | **0** |
| $P_2$ | **0** | 3 | 5 | 8 | **0** | **1** | **2** | **6** | **0** | 7 | 4 | 9 | **0** |

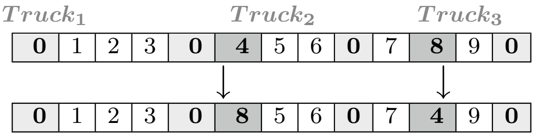
$$\begin{cases} 4 \leftrightarrow 1 \\ 5 \leftrightarrow 2 \\ 6 \leftrightarrow 6 \end{cases}$$

$$\Downarrow$$

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $O_1$ | **0** | 4 | 5 | 3 | **0** | **1** | **2** | **6** | **0** | 7 | 8 | 9 | **0** |
| $O_2$ | **0** | 3 | 2 | 8 | **0** | **4** | **5** | **6** | **0** | 7 | 1 | 9 | **0** |

**Fig. 3** Partially matched crossover (PMX)

**Fig. 4** Mutation operator



## 5 Computational experiments

The MCVRP mathematical model (1)–(12) is coded in IBM ILOG CPLEX 12.6 in order to output optimal or lower bound solutions. To explore further the solution space, we also coded the AVNS and GAPMX. All experiments were performed under a personal computer with *Intel*^ *Core*$^{TM}$ i5-5200U CPU @ 2.20 GHz 6,00 GO RAM and Windows 8.1 pro, 64-bit operating system, x64-based processor. Four sets of numerical experiments were performed, using the aforementioned four configurations in Table 1, in order to evaluate the performance of the AVNS. The problem size split into two classes:

- *Small, randomly-generated instances* Up to 15 customers, randomly generated according to the four configurations $G_1, \ldots, G_4$. We generated 18 instances for each configuration, giving rise to **72** small-sized problems.
- *Large instances.* Ranged between 50 and 200 customers. We used the **28** benchmark problems of Abdulkader et al. (2015).

### 5.1 Results for the small, randomly-generated instances

The design of the experimental strategy for small instances can be described in the following steps (see Fig. 5):

- Step 1: Solve the problem using CPLEX.
- Step 2: Perform the AVNS to get an optimal/ near optimal solution.
- Step 3: Evaluate AVNS solutions using the gap measurement.
- Step 4: Repeat the following until having a feasible gap.

Based on the four configurations displayed in Table 1, we drive an experimental investigation of the MCVRP with a number of customers $n \in \{5, 10, 15\}$. To do

**Table 1** Problem calibration for small MCVRP instances

| Inputs | | $G_1$ | $G_2$ | $G_3$ | $G_4$ |
|---|---|---|---|---|---|
| Customer $i_{(i=1,\ldots,\|V_c\|)}$ | $q_{ip}$ | [200, 600] | ]200, 600] | [600, 1000] | ]600, 1000] |
| Truck $k_{(k=1,\ldots,\|K\|)}$ | $Q_{max}$ | [3000, 3600] | [3600, 4200] | ]3000, 3600] | ]3600, 4200] |
| Rate $L(\%)$ | | [17, 123] | [21, 143] | [8, 64] | [10, 75] |

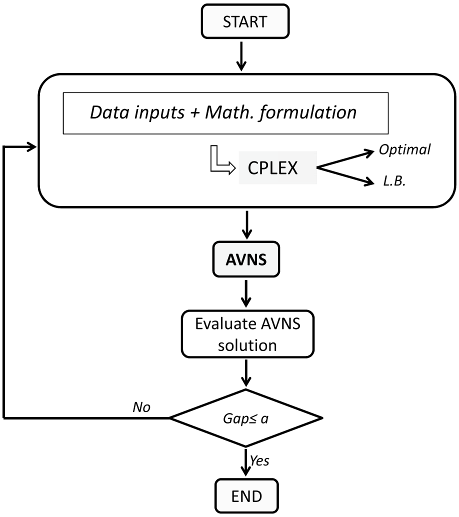


**Fig. 5** Main steps of the experimental strategy for small instances

so, 18 MCVRP instances for each configuration $G_{s(s=1,\ldots,4)}$ were randomly generated and used to assess the quality of the solutions produced by the proposed AVNS. We report in Table 1 the following problem parameters:

- Each *customer i* disposes of its demand $q_{ip}$ for each product $p$.
- Each *truck k* is characterized by:
  - $Q_{max}$: its capacity limit
  - $Q_p$: its capacity limit of each compartment
  - $|P|$: number of products (which is equal to the number of compartments)
  - $d_{ij}$: driving distance between customers $i$ and $j$
- $L = \frac{Q_{max}}{\sum_{i\in V_c}\sum_{p\in P} q_{ip}} \times 100$. It informs about the average quantity to be loaded in each truck for the delivery process. It is inversely proportional to the number of trucks.
- *ARPD* (Average Relative Percentage Deviation): the percentage between the best solution $z(x^*)$ obtained using CPLEX solver and the best found solution $z(x)$ obtained by the AVNS.

$$ARPD = \frac{z(x) - z(x^*)}{z(x^*)} \times 100 \tag{13}$$

As the problem calibration is expressed in terms of customers and trucks data, Table 1 outputs four configurations of the problem inputs as follows:

*Configurations* We used the CPLEX to find the optimal solutions for the small instances which can be used to evaluate the accuracy of the proposed AVNS.

Tables 2, 3, 4 and 5 report the numerical results related to the MCVRP tested regarding each problem configuration and display the ARPD between $z(x^*)$ the CPLEX solution and $z(x)$ the best value obtained by the AVNS.

*Configuration* $G_1$ Table 2 shows that CPLEX generates optimal solutions for all instances in set 1 ($C_{01} - C_{06}$) within an average runtime of 1.43 seconds and all instances in set 2 ($C_{07} - C_{12}$) in an average of 2.11 seconds.

However, half of the instances in set 3 ($C_{13} - C_{18}$) are solved optimally, and the require mean CPU of 313.35 seconds.

*Configuration* $G_2$ In Table 3, CPLEX is able to solve instances ($C_{19} - C_{24}$) of set 1 and ($C_{25} - C_{30}$) of set 2 to optimality in the average of 1.30 and 1.87 seconds, respectively, and five out of six instances in the last set of the instances ($C_{31} - C_{36}$) in average of approximately 1 hour. For the unsolved instance $C_{34}$, CPLEX ran for 3 hours and didn't converge.

*Configurations* $G_3$ *and* $G_4$ When observing Tables 4 and 5, all instances in these tables are solved optimally by CPLEX. The average runtime for the instances in Table 4 is 2.17 seconds and 2.26 seconds for the instances in Table 5.

It can be noticed that instances look harder to solve when the loading of orders level decreases and the number of trips' trucks increases, the run time of CPLEX increases quickly and some instances cannot be solved.

*Lower bound method* For the instances ($C_{13}$, $C_{16}$, $C_{18}$ and $C_{34}$), CPLEX reports an out of memory condition. Therefore, we tried to apply the the Lower Bound (LB) obtained from the continuous relaxation of the mathematical model for MCVRP to solve these instances using CPLEX. For this purpose, we relaxed the integrality requirement on the decision variable. This implies changing the integrality constraint $x_{ijk} \in \{0, 1\}$ by its continuous relaxation $0 \le x_{ijk} \le 1$.

Regarding the solution quality, results in Tables 2, 3, 4 and 5 indicate that our algorithm finds very close to optimal solutions to all instances solved by CPLEX with a substantially much shorter computation time. CPLEX cannot provide a feasible solution by applying the LB method to four instances ($C_{13}$, $C_{16}$, $C_{18}$ and $C_{34}$). The previous results (optimal solutions) confirm that the solutions obtained by our algorithm are much better than those obtained by applying the LB method. Moreover, the average computation time required by our proposed algorithm to solve all problem instances is much less than that required by CPLEX.

*Loading level* Obviously, an important factor that influences the difficulty of a MCVRP instances is the loading level $L$. In fact, this rate describes the ratio between the truck's capacity and total demand of customers.

$L$ appears to be interesting to evaluate in the context of the study, since it might be possible to determine the number of trucks needed for satisfying all customers' demands and, thus, the loading level of each truck for delivery of products to customers.

In this way, this rate influences the total driving distance $z(x^*)$ required to service all customers. When $L$ increases, the total travel distance decreases in the four configurations $G_1$, ..., $G_4$ (see Figure 6).

We have also observed that this rate is varied from one configuration to another. For example, in the first two configurations $G_1$ and $G_2$ the customers' demands $q_{ip}$ are the same in the problem $C_{01}$ and $C_{19}$ instances however the capacity limit of

**Table 2** Computational results for small-sized MCVRP with configuration G1

| Inst | $n$ | $L$ (%) | $Q_{max}$ | $Q_p$ | $\|P\|$ | CPLEX | | | AVNS | | ARPD |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | $z(x^*)$ | Type | CPU | $z(x)$ | CPU | |
| $C_{01}$ | 5 | 100.53 | 3000 | 1500 | 2 | 1473.651 | Opt. | 01.29 | 1473.65 | 0.001 | 0.00 |
| $C_{02}$ | | 77.65 | | 1000 | 3 | 3752.928 | Opt. | 01.40 | 3752.92 | 0.001 | 0.00 |
| $C_{03}$ | | 98.83 | 3300 | 1650 | 2 | 1473.651 | Opt. | 01.23 | 1473.65 | 0.001 | 0.00 |
| $C_{04}$ | | 81.28 | | 1100 | 3 | 377.690 | Opt. | 01.01 | 377.69 | 0.002 | 0.00 |
| $C_{05}$ | | 122.07 | 3600 | 1800 | 2 | 1432.752 | Opt. | 00.95 | 1432.75 | 0.001 | 0.00 |
| $C_{06}$ | | 73.57 | | 1200 | 3 | 377.690 | Opt. | 01.29 | 377.69 | 0.002 | 0.00 |
| $C_7$ | 10 | 44.05 | 3000 | 1500 | 2 | 3387.493 | Opt. | 02.21 | 3387.49 | 0.01 | 0.00 |
| $C_8$ | | 27.31 | | 1000 | 3 | 3126.957 | Opt. | 01.47 | 3126.95 | 0.02 | 0.00 |
| $C_9$ | | 46.49 | 3300 | 1650 | 2 | 3228.015 | Opt. | 01.36 | 3228.01 | 0.01 | 0.00 |
| $C_{10}$ | | 29.44 | | 1100 | 3 | 2890.640 | Opt. | 01.47 | 2890.64 | 0.01 | 0.00 |
| $C_{11}$ | | 50.43 | 3600 | 1800 | 2 | 4534.199 | Opt. | 01.45 | 4534.19 | 0.02 | 0.00 |
| $C_{12}$ | | 32.32 | | 1200 | 3 | 1352.962 | Opt. | 02.60 | 1352.96 | 0.01 | 0.00 |
| $C_{13}$ | 15 | 25.49 | 3000 | 1500 | 2 | 1170.375 | LB | 02.91 | 5418.63 | 08.05 | 362.97 |
| $C_{14}$ | | 17.72 | | 1000 | 3 | 8150.500 | Opt. | 02.59 | 8150.51 | 06.02 | 0.00 |
| $C_{15}$ | | 29.24 | 3300 | 1650 | 2 | 6216.832 | Opt. | 05.20 | 6216.81 | 03.31 | 0.00 |
| $C_{16}$ | | 18.74 | | 1100 | 3 | 3759.186 | LB | 25.30 | 4239.60 | 04.04 | 12.77 |
| $C_{17}$ | | 31.60 | 3600 | 1800 | 2 | 5124.037 | Opt. | 932.28 | 5124.03 | 05.68 | 0.00 |
| $C_{18}$ | | 22.01 | | 1200 | 3 | 5100.490 | LB | 04.36 | 5833.30 | 04.54 | 14.36 |
| *Average* | | 51.60 | | | | 3162.78 | | 55.02 | 3692.07 | 01.76 | 21.67 |

trucks $Q_{max} = 3600$ in $C_{19}$ greater than $Q_{max} = 3000$ in $C_{01}$. We can observe that the total driving distance $z(x^*)$ decreases when the capacity of trucks $Q_{max}$ or the rate $L$ increases.

Interestingly, it has been found that although the rate $L$ have an impact on the travel distance in the different configurations, and thus, it has also an influence on the average computing time needed to obtain the best solution.

When $L$ decreases, the problem instances become more difficult and costly in term of running time to solve. Therefore, it can be concluded that, the complexity of these problems dependent on this rate.

In addition, for the small-instances, the computational time of the CPLEX increases nearly linearly when the number of customer is under 15. However, the computational time grows factorially when the number of customers increases.

Based on the results presented in Tables 2, 3, 4 and 5, the configuration $G_3$ is solved with the shortest run time in an average of 2.18 seconds.

Besides to that, we can notice from Fig. 7 that the AVNS outperforms the CPLEX in terms of CPU time for all random generated instances (Table 6).

**Table 3** Computational results for small-sized MCVRP with configuration G2

| Inst | $n$ | $L$ (%) | $Q_{max}$ | $Q_p$ | $\|P\|$ | CPLEX | | | AVNS | | ARPD |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | $z(x^*)$ | Type | CPU | $z(x)$ | CPU | |
| $C_{19}$ | 5 | 120.64 | 3600 | 1800 | 2 | 1432.752 | Opt. | 01.48 | 1432.75 | 0.001 | 0.00 |
| $C_{20}$ | | 93.19 | | 1200 | 3 | 3752.928 | Opt. | 00.98 | 3752.92 | 0.001 | 0.00 |
| $C_{21}$ | | 116.80 | 3900 | 1950 | 2 | 1432.752 | Opt. | 01.40 | 1432.75 | 0.001 | 0.00 |
| $C_{22}$ | | 88.66 | | 1300 | 3 | 377.690 | Opt. | 01.38 | 377.69 | 0.002 | 0.00 |
| $C_{23}$ | | 142.42 | 4200 | 2100 | 2 | 1432.752 | Opt. | 01.44 | 1432.75 | 0.001 | 0.00 |
| $C_{24}$ | | 85.83 | | 1400 | 3 | 377.690 | Opt. | 01.12 | 377.69 | 0.002 | 0.00 |
| $C_{25}$ | 10 | 52.86 | 3600 | 1800 | 2 | 2899.866 | Opt. | 01.84 | 2899.86 | 0.01 | 0.00 |
| $C_{26}$ | | 32.78 | | 1200 | 3 | 2890.494 | Opt. | 02.08 | 2890.49 | 0.02 | 0.00 |
| $C_{27}$ | | 54.94 | 3900 | 1950 | 2 | 2899.866 | Opt. | 01.89 | 2899.86 | 0.02 | 0.00 |
| $C_{28}$ | | 34.79 | | 1300 | 3 | 2729.911 | Opt. | 01.72 | 2729.91 | 0.01 | 0.00 |
| $C_{29}$ | | 58.84 | 4200 | 2100 | 2 | 4148.503 | Opt. | 01.99 | 4148.50 | 0.01 | 0.00 |
| $C_{30}$ | | 37.71 | | 1400 | 3 | 1352.962 | Opt. | 01.75 | 1352.96 | 0.02 | 0.00 |
| $C_{31}$ | 15 | 30.59 | 3600 | 1800 | 2 | 2977.310 | Opt. | 4538.84 | 2977.31 | 02.40 | 0.00 |
| $C_{32}$ | | 21.27 | | 1200 | 3 | 7756.153 | Opt. | 531.01 | 7756.15 | 02.50 | 0.00 |
| $C_{33}$ | | 34.56 | 3900 | 1950 | 2 | 5555.044 | Opt. | 120.86 | 5555.04 | 03.10 | 0.00 |
| $C_{34}$ | | 22.14 | | 1300 | 3 | 3018.647 | LB | 21.72 | 4124.82 | 05.20 | 30.01 |
| $C_{35}$ | | 36.87 | 4200 | 2100 | 2 | 4490.106 | Opt. | 13472.35 | 4490.10 | 04.30 | 0.00 |
| $C_{36}$ | | 25.68 | | 1400 | 3 | 5839.038 | Opt. | 11596.08 | 5839.03 | 03.40 | 0.00 |
| *Average* | | 60.59 | | | | 3075.80 | | 1683.33 | 3137.26 | 01.17 | 1.67 |

## 5.2 Comparison with state of the art approaches

The used MCVRP benchmark instances are proposed by Abdulkader et al. (2015). They consist in an enrichment of Christofides' Christofides and Mingozzi (1979) benchmark using the strategy of Reed et al. (2014). The capacity ratio 1/3: the capacity of the vehicle is split into two compartments ($0.25Q$ and $0.75Q$).

Based on the results reported in Table 7, we conclude that: the *GAPMX* provides promising results as it finds the best-known for two instances and it outperforms the state-of-the-art approaches for four instances. The underline and bold results in Tables 7 and 8 denote the instances where our approach outperforms the state-of-the-art results.

We report in Table 7 the solution generated by the AVNS and GAPMX, and compared them to the ACS and HAC. The testbed counts **28** instances. Based on Table 7, we notice that:

- AVNS is the best in 23 instances: It outperforms the ACS and HAC in 11 instances and finds the best-known (Hits) for 12 instances (blue and red parts in Fig. 6).
- The average ARPD of all instances is around −0.9%.

**Table 4** Computational results for small-sized MCVRP with configuration G3

| Inst | $n$ | $L$ (%) | $Q_{max}$ | $Q_p$ | $\|P\|$ | CPLEX | | | AVNS | | ARPD |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | $z(x^*)$ | Type | CPU | $z(x)$ | CPU | |
| $C_{37}$ | 5 | 46.98 | 3000 | 1500 | 2 | 2687.079 | Opt. | 01.45 | 2687.07 | 0.001 | 0.00 |
| $C_{38}$ | | 31.54 | | 1000 | 3 | 5167.142 | Opt. | 01.42 | 5167.14 | 0.001 | 0.00 |
| $C_{39}$ | | 51.53 | 3300 | 1650 | 2 | 2417.556 | Opt. | 01.51 | 2417.55 | 0.002 | 0.00 |
| $C_{40}$ | | 34.82 | | 1100 | 3 | 570.535 | Opt. | 01.30 | 570.53 | 0.002 | 0.00 |
| $C_{41}$ | | 63.71 | 3600 | 1800 | 2 | 1473.651 | Opt. | 00.93 | 1473.65 | 0.001 | 0.00 |
| $C_{42}$ | | 39.92 | | 1200 | 3 | 570.535 | Opt. | 01.74 | 570.53 | 0.002 | 0.00 |
| $C_{43}$ | 10 | 21.57 | 3000 | 1500 | 2 | 4697.014 | Opt. | 01.31 | 4697.01 | 0.01 | 0.00 |
| $C_{44}$ | | 13.82 | | 1000 | 3 | 5040.698 | Opt. | 01.38 | 5040.69 | 0.01 | 0.00 |
| $C_{45}$ | | 22.97 | 3300 | 1650 | 2 | 4090.928 | Opt. | 01.39 | 4090.92 | 0.02 | 0.00 |
| $C_{46}$ | | 15.24 | | 1100 | 3 | 5040.698 | Opt. | 01.33 | 5040.69 | 0.02 | 0.00 |
| $C_{47}$ | | 25.53 | 3600 | 1800 | 2 | 5980.166 | Opt. | 02.24 | 5980.16 | 0.02 | 0.00 |
| $C_{48}$ | | 16.59 | | 1200 | 3 | 2427.651 | Opt. | 01.65 | 2427.65 | 0.02 | 0.00 |
| $C_{49}$ | 15 | 13.74 | 3000 | 1500 | 2 | 7229.157 | Opt. | 02.19 | 7229.15 | 03.25 | 0.00 |
| $C_{50}$ | | 08.93 | | 1000 | 3 | 12890.805 | Opt. | 01.19 | 12890.80 | 04.55 | 0.00 |
| $C_{51}$ | | 15.03 | 3300 | 1650 | 2 | 10095.044 | Opt. | 03.15 | 10095.04 | 03.50 | 0.00 |
| $C_{52}$ | | 09.72 | | 1100 | 3 | 8554.256 | Opt. | 01.96 | 8554.25 | 04.20 | 0.00 |
| $C_{53}$ | | 16.25 | 3600 | 1800 | 2 | 6822.633 | Opt. | 11.17 | 6822.63 | 04.50 | 0.00 |
| $C_{54}$ | | 10.75 | | 1200 | 3 | 10028.661 | Opt. | 01.87 | 10028.66 | 03.20 | 0.00 |
| *Average* | | 25.48 | | | | 5321.34 | | 02.18 | 5321.34 | 01.29 | 0.00 |

- For this set of instances (*vrpnc*1*b*, *vrpnc*5*a*, *vrpnc*5*b*, *vrpnc*6*b*, *vrpnc*9*a*, *vrpnc*9*b*, *vrpnc*10*a*, *vrpnc*10*b*, *vrpnc*12*a*, *vrpnc*12*b*, *vrpnc*13*a*, and *vrpnc*13*b*) the average of improvement is around −2.3%.
- For the rest, the average of the gap of the sub-optimal results is 0.16%.
- The best improvement is −9.7%.
- The worst gap is 0.8%.

In Table 8, we compare the relative deviation of the AVNS against the ITS while considering the HAC the best known solutions. We can notice from Table 8 that AVNS outperforms ITS approach in 6 instances. The radial deviation (RD) arranged in interval $[-9.6, 0.8]$.

### 5.3 Main remarks

Based on the values of the quality indicators which are the ARPD, gap and improvement, we observed:

First, for small-size instances, it can clearly be seen that the AVNS finds hits for 100% of all instances compared to the optimal solutions obtained through CPLEX, with an average ARPD of 0. In addition, CPLEX finds optimally 94.44% of instances, whereas 5.55% of these instances solves by using the LB method.

**Table 5** Computational results for small-sized MCVRP with configuration G4

| Inst | $n$ | $L$ (%) | $Q_{max}$ | $Q_p$ | $\|P\|$ | CPLEX | | | AVNS | | ARPD |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | $z(x^*)$ | Type | CPU | $z(x)$ | CPU | |
| $C_{55}$ | 5 | 56.38 | 3600 | 1800 | 2 | 1473.651 | Opt. | 01.63 | 1473.65 | 0.001 | 0.00 |
| $C_{56}$ | | 37.85 | | 1200 | 3 | 5167.142 | Opt. | 01.26 | 5167.14 | 0.001 | 0.00 |
| $C_{57}$ | | 60.90 | 3900 | 1950 | 2 | 1473.651 | Opt. | 01.57 | 1473.65 | 0.001 | 0.00 |
| $C_{58}$ | | 41.16 | | 1300 | 3 | 570.535 | Opt. | 01.29 | 570.53 | 0.001 | 0.00 |
| $C_{59}$ | | 74.33 | 4200 | 2100 | 2 | 1473.651 | Opt. | 01.28 | 1473.65 | 0.002 | 0.00 |
| $C_{60}$ | | 46.58 | | 1400 | 3 | 570.535 | Opt. | 01.74 | 570.53 | 0.001 | 0.00 |
| $C_{61}$ | 10 | 25.88 | 3600 | 1800 | 2 | 3812.303 | Opt. | 01.73 | 3812.30 | 0.01 | 0.00 |
| $C_{62}$ | | 16.59 | | 1200 | 3 | 5040.698 | Opt. | 01.13 | 5040.69 | 0.01 | 0.00 |
| $C_{63}$ | | 27.14 | 3900 | 1950 | 2 | 3812.303 | Opt. | 01.99 | 3812.30 | 0.01 | 0.00 |
| $C_{64}$ | | 18.01 | | 1300 | 3 | 5040.698 | Opt. | 01.34 | 5040.69 | 0.02 | 0.00 |
| $C_{65}$ | | 29.79 | 4200 | 2100 | 2 | 5980.166 | Opt. | 02.27 | 5980.16 | 0.01 | 0.00 |
| $C_{66}$ | | 19.36 | | 1400 | 3 | 2427.651 | Opt. | 01.00 | 2427.65 | 0.01 | 0.00 |
| $C_{67}$ | 15 | 16.49 | 3600 | 1800 | 2 | 4673.294 | Opt. | 15.90 | 4673.29 | 07.02 | 0.00 |
| $C_{68}$ | | 10.72 | | 1200 | 3 | 2343.023 | Opt. | 01.54 | 2343.02 | 03.35 | 0.00 |
| $C_{69}$ | | 17.76 | 3900 | 1950 | 2 | 2343.023 | Opt. | 01.34 | 2343.02 | 04.05 | 0.00 |
| $C_{70}$ | | 11.48 | | 1300 | 3 | 8554.256 | Opt. | 01.44 | 792.33 | 03.02 | 0.00 |
| $C_{71}$ | | 18.96 | 4200 | 2100 | 2 | 3257.882 | Opt. | 01.05 | 3257.88 | 04.20 | 0.00 |
| $C_{72}$ | | 12.55 | | 1400 | 3 | 3257.882 | Opt. | 01.29 | 3257.88 | 03.10 | 0.00 |
| *Average* | | 30.11 | | | | 2972.80 | | 02.27 | 2972.80 | 01.38 | 0.00 |

Second, for large-size instances, the AVNS greatly improves the quality of solutions against the state-of-the-art approaches in 12 instances with average improvement – 2.6% that is between – 0.06 and – 9.7%. Also, the AVNS finds hits for 40% of the benchmark instances. The average gap is about 0.8%, over all instances, that is between 0.02 and 1.7%.

It can be noted also that the GAPMX improves the best known results for 4 instances with an average – 0.3% that is between – 0.2 and – 0.5%.

## 5.4 Comparison with the one-compartment VRP

We conducted in the previous section an empirical investigation of the testbed related to 72 small-sized instances and 24 large-sized instances. To further highlight the strength of the proposed AVNS, we conducted a comparative study of the MCVRP with the one-compartment VRP (OCVRP) using the same instances. We provided a mathematical formulation and solved it using CPLEX for the

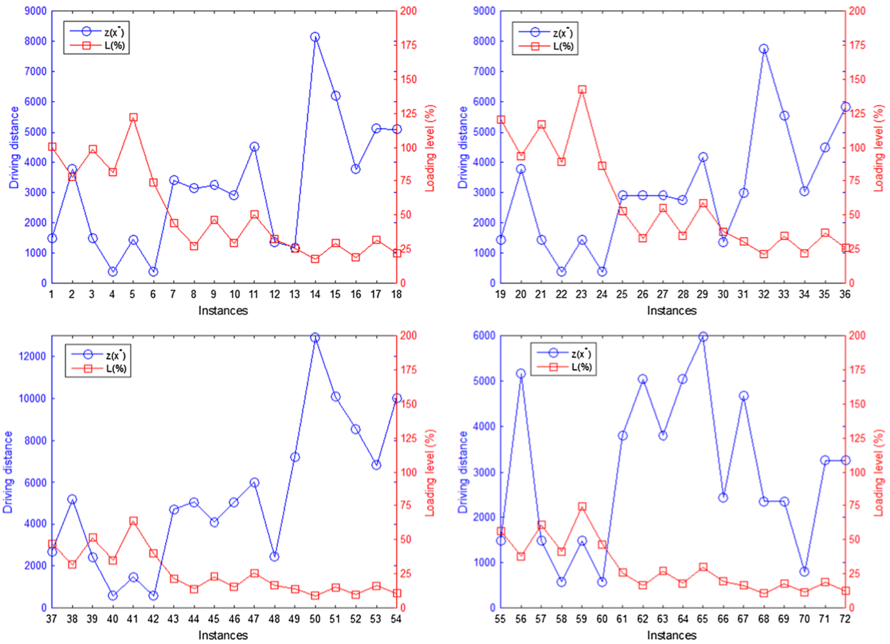


**Fig. 6** The trend of driving distance when the rate *L* changes

OCVRP. The mathematical formulation for the OCVRP considered in this paper is not available in the literature.

The only difference between the two VRP variants consists in splitting trucks' capacities for the MCVRP while keeping the whole capacity to one product for the OCVRP. As we succeeded, in most cases, to solve small instances to optimality, we propose to evaluate the convergence of OCVRP using the same dataset (see Table 6). We applied CPLEX for all the whole dataset. However, we were not able to generate the optimal solutions. That is why we displayed LBs, if not, we performed the AVNS (denoted by UB) to get a near optimal solution. We state in what follows the mathematical formulation of the OCVRP and used CPLEX and AVNS to point out the benefits of considering a multi-compartment version of the VRP. It can clearly be seen that the multi-compartment version of the VRP solves optimally 83.33, whereas 16.66% of instances solve by using the LB method of the four set of instances in the four configurations.

For the one-compartment version of the VRP, CPLEX finds 75% of the instances by using the LB method and the rest of these instances solve using the proposed AVNS.

Therefore, it can be concluded that, the one-compartment version of the VRP is more complex than the multi-compartment version of the VRP. The experiments show that the multi-compartment version of the VRP converges quickly against the one-compartment version.

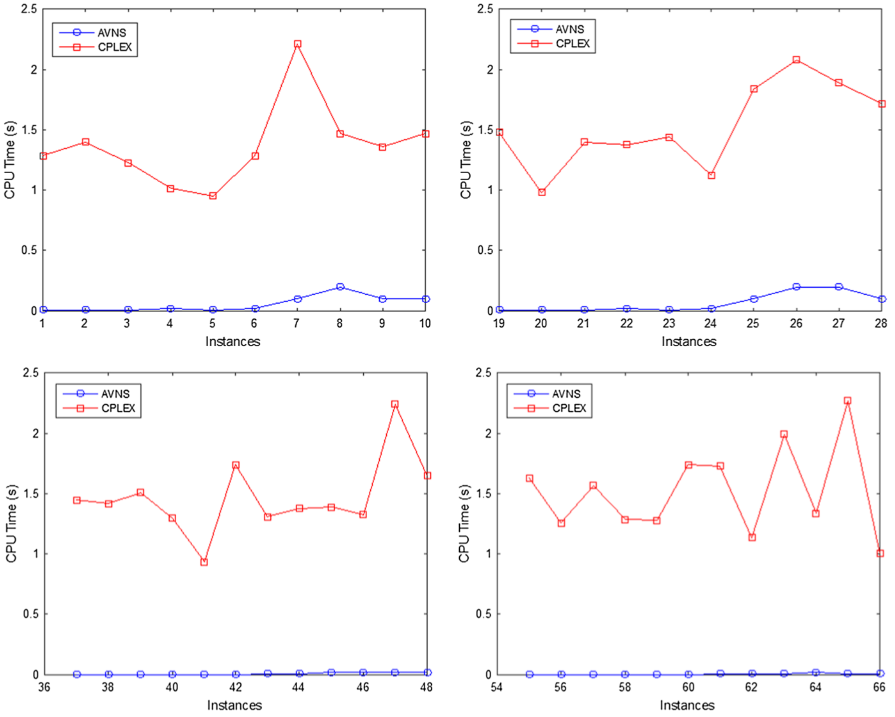


**Fig. 7** CPLEX versus AVNS in terms of CPU Time

The one-compartment VRP

| | | | |
|---|---|---|---|
| *MinS.t.* | $z(x) = \sum_{i\in V}\sum_{j\in V, j\neq i}\sum_{k\in K} d_{ij}\, x_{ijk}$ | | (1) |
| | $\sum_{k\in K} y_{ik} = \lvert P\rvert$ | $i \in V_c$ | (2) |
| | $\sum_{k\in K} y_{0k} \leq \lvert K\rvert$ | | (3) |
| | $\sum_{i\in V, i\neq j} x_{ijk} = y_{jk}$ | $j \in V_c, k \in K$ | (4) |
| | $\sum_{j\in V, j\neq i} x_{ijk} = y_{ik}$ | $i \in V_c, k \in K$ | (5) |
| | $u_{ik} - u_{jk} + Q_{max}\, x_{ijk} \leq Q_{max} - q_{ip}$ | $i,j \in V_c, i \neq j, k \in K, p \in P$ | (6) |
| | $q_{ip} \leq u_{ik} \leq Q_{max}$ | $i \in V_c, k \in K, p \in P$ | (7) |
| | $z_{jkp} \leq \sum_{i\in V, i\neq j} x_{ijk}$ | $j \in V_c, k \in K, p \in P$ | (8) |
| | $\sum_{k\in K} z_{jkp} = 1$ | $j \in V_c, p \in P$ | (9) |
| | $\sum_{j\in V_c} z_{jkp}\, q_{jp} \leq Q_{max}$ | $k \in K, p \in P$ | (10) |
| | $\sum_{p\in P} w_{kp} \leq 1$ | $k \in K$ | (11) |
| | $z_{jkp} \leq w_{kp}$ | $j \in V_c, k \in K, p \in P$ | (12) |
| | $x_{ijk}, z_{jkp}, y_{ik}, w_{kp} \in \{0,1\}$ | $i,j \in V, k \in K,\ p \in P$ | (13) |

Objective function: minimize the travel distance

Constraints: Equality between the number of trucks and the number of requested products, subtour elimination, each ordered product is served by one truck, trucks capacity constraints

**Table 6** Comparison between MCVRP and OCVRP for the small instances

| Inst | MCVRP | | | | OCVRP | | | |
|---|---|---|---|---|---|---|---|---|
| | $z(x^*)$ | $\|K\|$ | Type | CPU | $z(x^{**})$ | $\|K\|$ | Type | CPU |
| $C_{13}$ | 1170.375 | 5 | LB | 02.91 | 1633.521 | 6 | LB | 08.87 |
| $C_{14}$ | 8150.500 | 6 | Opt. | 02.59 | 8051.557 | 6 | LB | 03.96 |
| $C_{15}$ | 6216.832 | 4 | Opt. | 239.31 | 5370.234 | 5 | LB | 06.46 |
| $C_{16}$ | 3759.186 | 6 | LB | 25.30 | 2533.781 | 6 | LB | 14.55 |
| $C_{17}$ | 5124.037 | 4 | Opt. | 932.28 | 6652.395 | 7 | LB | 451.10 |
| $C_{18}$ | 5100.490 | 5 | LB | 04.36 | 4623.100 | 7 | UB | 5.30 |
| $C_{31}$ | 2977.310 | 4 | Opt. | 4538.84 | 1615.579 | 6 | LB | 10.08 |
| $C_{32}$ | 7756.153 | 5 | Opt. | 531.01 | 7881.143 | 6 | LB | 09.36 |
| $C_{33}$ | 5555.044 | 4 | Opt. | 120.86 | 5216.805 | 6 | LB | 16.70 |
| $C_{34}$ | 3018.647 | 5 | LB | 21.72 | 2423.790 | 6 | LB | 07.70 |
| $C_{35}$ | 4490.106 | 3 | Opt. | 13472.35 | 6632.877 | 6 | LB | 10.88 |
| $C_{36}$ | 5839.038 | 5 | Opt. | 11596.08 | 9950.045 | 6 | LB | 08.29 |
| $C_{49}$ | 7229.157 | 8 | Opt. | 02.19 | 7684.640 | 9 | UB | 06.20 |
| $C_{50}$ | 12890.805 | 12 | Opt. | 01.19 | 14025.230 | 12 | UB | 07.25 |
| $C_{51}$ | 10095.044 | 7 | Opt. | 03.15 | 6554.919 | 12 | LB | 4259.68 |
| $C_{52}$ | 8554.256 | 11 | Opt. | 01.96 | 9853.360 | 12 | UB | 05.30 |
| $C_{53}$ | 6822.633 | 7 | Opt. | 11.17 | 6810.719 | 8 | LB | 17.23 |
| $C_{54}$ | 10028.661 | 10 | Opt. | 01.87 | 10258.220 | 12 | LB | 60.92 |
| $C_{67}$ | 4673.294 | 7 | Opt. | 15.90 | 1735.323 | 8 | LB | 960.81 |
| $C_{68}$ | 2343.023 | 10 | Opt. | 01.54 | 3451.340 | 12 | UB | 06.70 |
| $C_{69}$ | 2343.023 | 6 | Opt. | 01.34 | 6401.053 | 6 | LB | 17.12 |
| $C_{70}$ | 8554.256 | 9 | Opt. | 01.44 | 9652.370 | 12 | UB | 07.20 |
| $C_{71}$ | 3257.882 | 6 | Opt. | 01.05 | 6931.546 | 7 | LB | 38.38 |
| $C_{72}$ | 3257.882 | 9 | Opt. | 01.29 | 10436.593 | 9 | LB | 2386.43 |
| *Average* | 5608.170 | | | 1432.554 | 6667.957 | | | 377.892 |

We explained in more details this mathematical model as follows:

- The objective function (1) minimizes the total travel distance of all used trucks.
- Constraint (2) ensures that the number of trucks service each customer must be equal to the number of products.
- Constraint (3) imposes that subset of the trucks leave the depot.
- Constraints (4) and (5) force each truck to arrive and leave from customer $i$ only if it serves that customer.
- Constraints (6) and (7) are subtour elimination constraints imposing both the capacity and connectivity of the feasible routes.
- Constraint (8) couples routing variables $x_{ijk}$ to demand variables $z_{jkp}$.
- Constraint (9) means that each product ordered by a customer is delivered by exactly one truck.

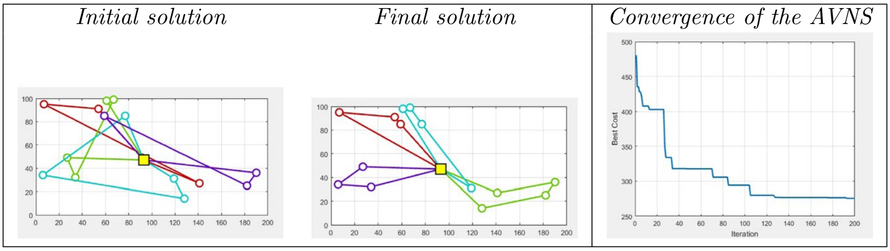
**Table 7** Computational results with stat-of-the-art metaheuristics on the MCVRP benchmark instances

| Inst | *n* | ACS | | HAC | | AVNS | | GAPMX | |
|---|---|---|---|---|---|---|---|---|---|
| | | Reed et al. (2014) | | Abdulkader et al. (2015) | | This paper | | This paper | |
| | | Cost | CPU (s) | Cost | CPU (s) | Cost | CPU (s) | Cost | CPU (s) |
| *vrpnc1a* | 50 | 569.564 | 16 | 550.70 | 5 | **550.7** | 4 | **551** | 20 |
| *vrpnc1b* | | 569.118 | 17 | 551.94 | 5 | **506,18** | 4 | **551** | 21 |
| *vrpnc2a* | 75 | 957.525 | 36 | 890.68 | 15 | **890.68** | 14 | 901 | 30 |
| *vrpnc2b* | | 954.856 | 35 | 918.96 | 14 | **918.96** | 14 | **914** | 31 |
| *vrpnc3a* | 100 | 964.132 | 122 | 874.07 | 40 | 880.843 | 22 | 894 | 112 |
| *vrpnc3b* | | 959.327 | 122 | 895.26 | 44 | 900.633 | 22 | 914 | 113 |
| *vrpnc4a* | 150 | 1253.86 | 345 | 1126.12 | 146 | **1126.12** | 104 | 1243 | 310 |
| *vrpnc4b* | | 1254.51 | 336 | 1159.48 | 151 | **1159.48** | 120 | 1204 | 340 |
| *vrpnc5a* | 199 | 1587.02 | 688 | 1444.29 | 257 | **1408.86** | 312 | 1470 | 690 |
| *vrpnc5b* | | 1640.59 | 676 | 1525.87 | 236 | **1515.25** | 308 | 1570 | 680 |
| *vrpnc6a* | 50 | 573.274 | 13 | 557.49 | 11 | **557.49** | 6 | 560 | 12 |
| *vrpnc6b* | | 573.378 | 13 | 559.37 | 10 | **505.56** | 8 | 563 | 11 |
| *vrpnc7a* | 75 | 997.007 | 33 | 928.24 | 28 | **928.24** | 30 | 980 | 30 |
| *vrpnc7b* | | 969.337 | 32 | 932.67 | 26 | **932.67** | 28 | 960 | 31 |
| *vrpnc8a* | 100 | 963.381 | 97 | 882.96 | 93 | 890.2 | 90 | **880** | 92 |
| *vrpnc8b* | | 976.212 | 97 | 884.85 | 95 | 889.3 | 105 | **884** | 92 |
| *vrpnc9a* | 150 | 1343.08 | 273 | 1228.88 | 326 | **1211.32** | 320 | **1221** | 280 |
| *vrpnc9b* | | 1346.63 | 274 | 1226.58 | 333 | **1220.58** | 340 | **1226** | 283 |
| *vrpnc10a* | 199 | 1645.58 | 606 | 1511.65 | 624 | **1505.23** | 602 | 1680 | 702 |
| *vrpnc10b* | | 1659.94 | 608 | 1526.02 | 620 | **1517.65** | 608 | 1683 | 680 |
| *vrpnc11a* | 120 | 1133.88 | 281 | 1110.45 | 75 | **1110.45** | 70 | 1130 | 210 |
| *vrpnc11b* | | 1247.49 | 280 | 1221.73 | 87 | **1221.73** | 77 | 1250 | 220 |
| *vrpnc12a* | 100 | 911.861 | 105 | 912.64 | 15 | **901.36** | 20 | 913 | 80 |
| *vrpnc12b* | | 970.833 | 100 | 950.79 | 30 | **936.25** | 22 | 960 | 90 |
| *vrpnc13a* | 120 | 1577.45 | 171 | 1556.46 | 117 | **1550.56** | 90 | 1570 | 180 |
| *vrpnc13b* | | 1572.11 | 168 | 1550.12 | 123 | **1540.37** | 98 | 1572 | 163 |
| *vrpnc14a* | 100 | 914.857 | 91 | 911.35 | 34 | **911.35** | 35 | 913 | 80 |
| *vrpnc14b* | | 970.933 | 91 | 965.84 | 38 | **965.84** | 40 | 973 | 83 |
| *Average* | | 1109.205 | 204.5 | 1048.41 | 128.5 | **1041.2** | **125.4** | 1102 | 180 |

- Constraint (10) checks the demand for each product delivered by each truck to customers is less than it capacity.
- Constraint (11) imposes that each truck is loaded by exactly one product.
- Constraint (12) ensures that each product may only be loaded in a truck if the corresponding customer is visited by this truck.
- Constraints (13) define the domains of decision variables.

**Table 8** Computational results with stat-of-the-art metaheuristics on the MCVRP benchmark instances

| Inst | *n* | HAC | | ITS | | AVNS | |
|---|---|---|---|---|---|---|---|
| | | Abdulkader et al. (2015) | | Silv. et al. (2017) | | This paper | |
| | | *vrpncXa* | *vrpncXb* | *vrpncXa* | *vrpncXb* | *vrpncXa* | *vrpncXb* |
| 1 | 50 | 0 | 0 | 0 | − 0.7 | **0** | **− 8.3** |
| 2 | 75 | 0 | 0 | − 2.1 | − 5.2 | 0 | 0 |
| 3 | 100 | 0 | 0 | − 0.6 | − 5.2 | 0.6 | 0.7 |
| 4 | 150 | 0 | 0 | − 2.4 | − 4.5 | 0 | 0 |
| 5 | 199 | 0 | 0 | − 3.5 | − 8.5 | − 2.4 | − 0.7 |
| 6 | 50 | 0 | 0 | 0 | − 0.7 | **0** | **− 9.6** |
| 7 | 75 | 0 | 0 | − 0.2 | − 0.1 | 0 | 0 |
| 8 | 100 | 0 | 0 | − 0.7 | − 1.1 | 0.8 | 0.5 |
| 9 | 150 | 0 | 0 | − 2.1 | − 2.2 | − 1.4 | − 0.4 |
| 10 | 199 | 0 | 0 | − 3.8 | − 3.4 | − 0.4 | − 0.3 |
| 11 | 120 | 0 | 0 | − 0.4 | − 2 | 0 | 0 |
| 12 | 100 | 0 | 0 | − 0.6 | 0.2 | **− 1.23** | **− 1.52** |
| 13 | 120 | 0 | 0 | − 0.7 | − 0.3 | − 0.4 | **− 0.6** |
| 14 | 100 | 0 | 0 | 0 | 0.2 | **0** | **0** |



# 6 Conclusion

This paper introduced the multi-compartment vehicle routing problem (MCVRP) that includes a set of trucks, a set of customers and a set of products. Each truck is equipped with multiple compartments, and each compartment is dedicated to a single product. The demand of each customer for all products should be fully satisfied. The delivery of the products related to a customer should be ensured by a single truck. The objective of the MCVRP includes the minimization of the total travel distance that incurs a minimization of the fleet size. Considering the *NP*-Hardness of the MCVRP, we proposed an AVNS and a GAPMX as approximate methods. The obtained near-optimal solutions are compared to existing metaheuristics. The AVNS reached optimal solutions obtained by CPLEX for all small-sized problems. For large instances, the numerical results also indicated that the AVNS outperformed the state-of-the-art approaches in 12 instances. In addition, the AVNS finds hits for

40% of the benchmark instances. The GAPMX improved the best known results for 4 instances with an average of -0.3. Through a comparative study of the AVNS and GAPMX against the state-of-the-art methods, the efficiency of the proposed algorithms is proved in terms of solution quality. In fact, our approaches generated new best-known solutions for the benchmark instances. A comparative study of the MCVRP with the OCVRP showed that the use of trucks with multiple compartments fastens the convergence of the optimizer in finding the optimal/near-optimal solution. The assumption that the demands of customers are introduced dynamically is an interesting suggestion for future research. The dynamic version of the MCVRP can be approached by a metaheuristic specifically designed to handle the online launching of demands.

## References

Abdulkader MM, Gajpal Y, ElMekkawy TY (2015) Hybridized ant colony algorithm for the multi compartment vehicle routing problem. Appl Soft Comput 37:196–203

Avella P, Boccia M, Sforza A (2004) Solving a fuel delivery problem by heuristic and exact approaches. Eur J Oper Res 152(1):170–179

Battarra M, Cordeau J-F, Iori M (2014) Pickup and delivery problems for goods transportation. In: Vehicle routing: problems, methods, and applications, second edition, Chap 6. ISBN: 978-1-61197-358-7

Breunig U, Schmid V, Hartl R, Vidal T (2016) A large neighbourhood based heuristic for two-echelon routing problems. Comput Oper Res 76:208–225

Brimberg J, Mladenović N, Todosijević R, sević D U (2017) Less is more: solving the max-mean diversity problem with variable neighborhood search. Inf Sci 382–383:179–200

Bula GA, Prodhon C, Gonzalez FA, Afsar HM, Velasco N (2017) Variable neighborhood search to solve the vehicle routing problem for hazardous materials transportation. J Hazard Mater 324(Part B):472–480

Chajakis ED, Guignard M (2003) Scheduling deliveries in vehicles with multiple compartments. J Glob Optim 26(1):43–78

Christofides PTN, Mingozzi A (1979) The vehicle routing problem, In: Combinatorial optimization. Wiley, Chichester, pp 315–338

Coelho LC, Laporte G (2015) Classification, models and exact algorithms for multi-compartment delivery problems. Eur J Oper Res 242(3):854–864

Cornillier F, Boctor F, Renaud J (2012) Heuristics for the multi-depot petrol station replenishment problem with time windows. Eur J Oper Res 220(2):361–369

Cornillier F, Boctor FF, Laporte G, Renaud J (2008) An exact algorithm for the petrol station replenishment problem. J Oper Res Soc 59(5):607–615

Cornillier F, Boctor FF, Laporte G, Renaud J (2008) A heuristic for the multi-period petrol station replenishment problem. Eur J Oper Res 191(2):295–305

Cornillier F, Laporte G, Boctor FF, Renaud J (2009) The petrol station replenishment problem with time windows. Comput Oper Res 36(3):919–935

Dirac P (1953) The lorentz transformation and absolute time. Physica 19(1–12):888–896

Elbek M, Wøhlk S (2016) A variable neighborhood search for the multi-period collection of recyclable materials. Eur J Oper Res 249(2):540–550

Fallahi AE, Prins C, Calvo RW (2008) A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. Comput Oper Res 35(5):1725–1741

Goldberg D, Lingle R (1985) Alleles, loci and the traveling salesman problem. In: Proceedings of the 1st international conference on genetic algorithms and their applications, Los Angeles, pp 154–159

Haddadene SRA, Labadie N, Prodhon C (2016) A grasp x ils for the vehicle routing problem with time windows, synchronization and precedence constraints. Expert Syst Appl 66:274–294

Hansen P, Mladenović N, Moreno Pérez J A (2008) Variable neighbourhood search: methods and applications. 4OR 6(4):319–360

Hansen P, Mladenović N, Pérez JAM (2008) Variable neighborhood search. Eur J Oper Res 191(3):593–595

Kaabachi I, Yahyaoui H, Dekdouk A, Krichen S (2016) A multi-operator genetic algorithm for the multi-compartments vehicle routing problem. In: The 6th international conference on information systems, logistics and supply chain, June 1–4, Bordeaux, France (Indexed by EI, INSPEC, DBLP and Thomson ISI)

Kumar N, Karambir RK (2012) A comparative analysis of pmx, cx and ox crossover operators for solving travelling salesman problem. Int J Latest Res Sci Technol 1:98–101

Lahyani R, Coelho LC, Khemakhem M, Laporte G, Semet F (2015) A multi-compartment vehicle routing problem arising in the collection of olive oil in tunisia. Omega 51:1–10

Li K, Tian H (2016) A two-level self-adaptive variable neighborhood search algorithm for the prize-collecting vehicle routing problem. Appl Soft Comput 43:469–479

Mladenović N, Hansen P (1997) Variable neighborhood search. Comput Oper Res 24(11):1097–1100

Muyldermans L, Pang G (2010) On the benefits of co-collection: experiments with a multi-compartment vehicle routing algorithm. Eur J Oper Res 206(1):93–103

Ng WL, Leung SCH, Lam JKP, Pan SW (2008) Petrol delivery tanker assignment and routing: a case study in hong kong. J Oper Res Soc 59(9):1191–1200

Oppen J, Løkketangen A (2008) A tabu search approach for the livestock collection problem. Comput Oper Res 35(10):3213–3229

Paredes-Belmar G, Marianov V, Bronfman A, Obreque C, Ler-Villagra A (2016) A milk collection problem with blending. Transp Res Part E: Logist Transp Rev 94:26–43

Popović D, Vidović M, Radivojević G (2012) Variable neighborhood search heuristic for the inventory routing problem in fuel delivery. Expert Syst Appl 39(18):13390–13398

Rahimian E, Akartunali K, Levine J (2017) A hybrid integer programming and variable neighbourhood search algorithm to solve nurse rostering problems. Eur J Oper Res 258(2):411–423

Reed M, Yiannakou A, Evering R (2014) An ant colony algorithm for the multi-compartment vehicle routing problem. Appl Soft Comput 15:169–176

Wang Y, Chen Y, Lin Y (2017) Memetic algorithm based on sequential variable neighborhood descent for the minmax multiple traveling salesman problem. Comput Ind Eng 106:105–122