Transportation Research Part C 70 (2016) 100–112

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

# A real-life Multi Depot Multi Period Vehicle Routing Problem with a Heterogeneous Fleet: Formulation and Adaptive Large Neighborhood Search based Matheuristic

Simona Mancini

*Politecnico di Torino, Italy*

A R T I C L E  I N F O

*Article history:*
Received 29 April 2014
Received in revised form 4 May 2015
Accepted 19 June 2015
Available online 7 July 2015

*Keywords:*
Multi depot vehicle routing
Heterogeneous fleet
Multi period
Large Neighborhood Search
Matheuristic

A B S T R A C T

In this paper, a new rich Vehicle Routing Problem that could arise in a real life context is introduced and formalized: the Multi Depot Multi Period Vehicle Routing Problem with a Heterogeneous Fleet. The goal of the problem is to minimize the total delivery cost. A heterogeneous fleet composed of vehicles with different capacity, characteristics (i.e. refrigerated vehicles) and hourly costs is considered. A limit on the maximum route duration is imposed. Unlike what happens in classical multi-depot VRP, not every customer may/will be served by all the vehicles or from all the depots. The planning horizon, as in most real life applications, consists of multiple periods, and the period in which each route is performed is a variable of the problem. The set of periods, within the time horizon, in which the delivery may be carried out is known for each customer. A Mixed Integer Programming (MIP) formulation for MDMPVRPHF is presented in this paper, and an Adaptive Large Neighborhood Search (ALNS) based Matheuristic approach is proposed, in which different destroy operators are defined. Computational results, pertaining to realistic instances, which show the effectiveness of the proposed method, are provided.

## 1. Introduction

Real-life freight distribution problems present a high degree of complexity, mostly because of the need to respect a variety of constraints concerning multi-dimensional vehicle capacity (weight and volume), characteristics of the vehicles (refrigerated vehicles for perishable food, exc.), route duration/length restriction, time windows, and product/product, product/vehicle and/or customer/vehicle compatibility, and they often require objective functions (which are more complex than the classical distance minimization) in which different factors must be taken into account.

Despite the great relevance of these problems in practical applications, they have received limited attention from an academic point of view. A pioneer contribution to this field can be found in Semet and Taillard (1993), where the authors deal with a heterogeneous fleet with customer accessibility restrictions. A unified heuristic framework for rich Vehicle Routing Problems has been proposed in Caric et al. (2008), while in Weise et al. (2009) and Confessore et al. (2008), an evolutionary approach has been used to address real-life routing problems. Industrial vehicle routing applications and the heuristic methods used to solve them have been reviewed in Hasle and Kloster (2007). Most of the papers addressing real-life routing problems are mainly focused on heuristic methods, while modeling and exact approaches are very rare in the literature. In Goel and Gruhn (2008), a General Vehicle Routing Problem has been formalized, and a mathematical formulation and a Reduced Variable Neighborhood Search procedure have been proposed.

http://dx.doi.org/10.1016/j.trc.2015.06.016
0968-090X/© 2015 Elsevier Ltd. All rights reserved.

Real-life Vehicle Routing Problems arise daily in a variety of different contexts and applications, and they usually introduce certain complications compared to the basic Vehicle Routing Problem (VRP) that has been addressed extensively in the literature. Most of the complications are related to the following aspects:

- *Planning horizon:* in the classical VRP, a single time period is addressed while, in real-life problems, routes are planned for a given planning horizon that may consist of multiple periods. A customer may be served only over a subset of these periods, Francis et al. (2008). Some customers may require to be served over more than one period and, in certain applications, the number of periods in which the customer is served in fact becomes a variable of the problem.
- *Customer:* in a basic VRP, each customer has a demand; in more complicated problems, the customers may have specific requirements related to the service time and/or the vehicle type. Different types of services, such as pickup, delivery or pickup-and-delivery services, could also exist, whose customers should be visited on the basis of their attractiveness. In some cases, split delivery is allowed, i.e. customers are allowed to be visited several times by several vehicles instead of just once by one vehicle, Archetti and Speranza (2008). In some applications on multiple period planning, the customers require deliveries every day and they can store products for the following days, if they have received more than what they need. In this case, the distributor needs to make a routing plan according to the demands and inventories of the customers.
- *Depot:* there can be multiple depots in a large distribution network, Mancini (2012); Perboli et al. (2011); Crainic et al. (2010, 2012, 2009); Gonzalez-Feliu (2012) to reduce the total costs of the supply chain. A customer may be served by all the depots or only by a subset.
- *Vehicle:* the vehicles used for delivery can have different capacities and sizes. There are usually a limited number of vehicles available in real-life planning. A vehicle may be used in multiple trips instead of a single trip in a routing plan. In multiple depot problems, each vehicle may be associated to a base depot, i.e. the vehicles must start from and end up at their base depot, or vehicles can be considered free to end their trip at another depot and restart from there in the following time period.
- *Driver:* in most real-life problems, distributors need to consider the drivers' working regulations, e.g. the working shift and break rules. In addition, specific qualifications may be required to drive particular vehicles (for instance: very large trucks). This means that not all the drivers may be allowed to drive all the vehicles.
- *Objective:* other than classical distances and/or travel time minimization, real-life problems may present rather complex objective functions, which may include the minimization of the difference between the longest and shortest routes to balance the workload among drivers, the minimization of the number of vehicles in order to save on large overheads, and/or the maximization of the number of served customers in order to improve the service level. In some applications, robustness of the solution is also an important issue. These different objectives may be in conflict with each other.
- *Uncertainties:* there can be uncertainties in the route planning. For example, the locations and/or the demands of customers are unknown at the beginning but are revealed over time when the vehicles have already started their trips. These kinds of problems are generally grouped under the name of Dynamic VRP, Larsen (2001) and Pillac et al. (2013). A dynamic version of the multi-period VRP has been proposed in Wen et al. (2010).

## 2. Problem description

The real-life problem that is the subject of this study is described in detail in this section. This research arose from the needs of a transport carrier to deal with the weekly planning of delivery operations for a company which uses several depots, where goods are stored and/or produced. There is a known fleet of vehicles, located at the depots, and a set of available drivers for which a fixed cost, in addition to a variable cost given by the driving time multiplied by a hourly driving cost, must be considered if they are scheduled according to a weekly plan.

This is a Multi-Period problem, because each customer must be served in one of the time slots (typically, but not necessarily, days), in which he/she is available. Furthermore, other constraints on the periods in which the delivery can be made may arise due to the fact that customers need to receive the goods within a given deadline, or the goods must be delivered after a given time slot, because they are not available before.

The problem is Multi-Depot, but unlike what happens in classical Multi-Depot Vehicle Routing Problems (MDVRP), Toth and Vigo (2002) in which routes must end at the same depot from which they started, we have considered routes that could end at different depots. This version of MDVRP, which has not received much interest by the academic community, is very common in practical applications. In fact, it could be more convenient to end a route in a different depot from the starting one, thus avoiding a long trip back to the depot.

This feature is frequently adopted by transportation companies that operate throughout a large territory and work with strict deadlines. A possible case study is reported in Fig. 1. The company holds three depots in the North of Italy, one in Turin (TU), one in Milan (MI) and the third in Venice (VE). The customers are partially from within the cities and their surrounding urban areas, while others are from areas between the cities. The customers marked in blue must be served on the first day of delivery, (D1), those in red on D2 and in green on D3. In many applications, such as in grocery deliveries, each customer area is visited once or twice a week, instead of every day. Imposing the constraint that vehicles must return to their starting depot would imply the need to use more vehicles to perform all the deliveries within the given deadline, due to the long trip back to the departure depot. Instead, if the vehicles were allowed to end their routes in another depot, a better exploitation of the

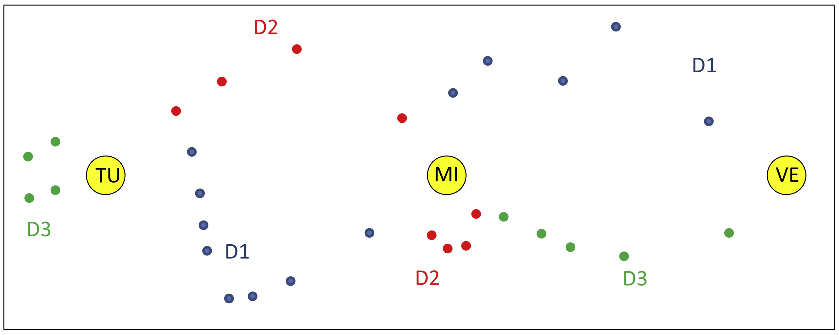

**Fig. 1.** Sample customer distribution.

available fleet would be obtained. A feasible solution to the problem, which involves only two vehicles, is reported in Fig. 2. If the constraint that each vehicle must return to its starting depot is introduced, the minimum number of vehicles necessary to perform the delivery operations increases to 4; (a feasible solution is reported in Fig. 3).

The routes are drawn in Figs. 2 and 3. Each route is identified by a keyword indicating the day in which the route is scheduled and the vehicle by which it is performed. (For example: keyword D1-V2 indicates that a route is performed on day 1 by vehicle 2). As can be noted from the figures, if the constraint that each vehicle must return to its starting depot is relaxed, the problem can be solved with two vehicles, one initially located in Turin and the other one in Venice, and introducing a total number of 6 routes. Instead, when the constraints are considered, 4 vehicles and 10 routes are necessary, other than a sensibly larger global route length. Therefore, in this case leaving the vehicles free to end their routes in any depot leads to a much better solution. On the other hand, the problem becomes more complicated to be heuristically solved. In fact, if we suppose that each vehicle has to return to its starting depot, the multi-period can be decomposed into N multi-depot problems, one for each period, which can be solved separately. If this constraint no longer holds, the decomposition is no longer possible, while the location of the vehicles at the depots at the beginning of a period depends on the activities exploited in the previous period. Therefore, all the periods must be considered together at the same time, and the final depot for each vehicle must be chosen taking into account the activity to be exploited in the following days.

A heterogeneous fleet composed of vehicles characterized by a different capacity (expressed in terms of loading units), and a different cost per km has been considered, as found for most of the VRPs with a heterogeneous fleet in the literature (Salhi et al., 2014; Brandao, 2011; Choi and Tcha, 2007) in which a fixed but different usage cost is considered for each vehicle class. For a complete survey of VRPs with a heterogeneous fleet, the readers may refer to Baldacci et al. (2008) and the references therein. Furthermore, it has been considered that the vehicles may have different characteristics (for instance, refrigerated vehicles) which can be required by some customers but must be avoided for others. This situation is frequently found for food deliveries, in which perishable products require refrigerated vehicles. If it is supposed that the average speed is known, it is easy to transform the cost per km to a cost per hour, in order to make it comparable with the driving cost. The location of each vehicle at the beginning of the analyzed time-horizon is supposed to be known in advance. Not all the customers may be served by all the vehicles and from all the depots. This restriction may be due to the specific requirements of the product (for instance, refrigerated vehicles for perishable food, freezer vehicles for ice cream and frozen foods), or to the customer's location (for instance, customers located in the city center cannot be served by very large vehicles as they are not allowed to enter that zone). Restrictions on the depots from which customers may be served hold when the products requested by the customer are not available at each depot. Restrictions on the maximum route duration due to limitations on the drivers' working hour limitations are also considered.

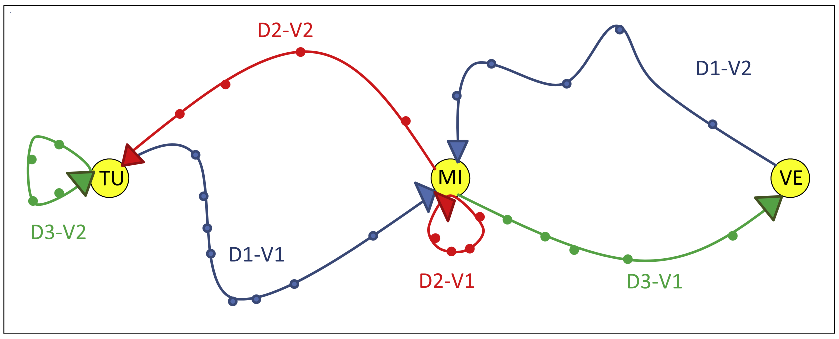

**Fig. 2.** Solution of the Multi-Period Multi-Depot without the constraint of a vehicle having to return to the starting depot.

### 2.1. Problem definition

The MDMPVRPHF is formally defined in this subsection. The problem consists in serving a set of customers, $I$, at the minimum cost. Each customer $i$ requires a quantity of goods $q_i$ which can be delivered from one of several depots $d$ belonging to the set of depots $D$, depending on the availability of the requested products at the depots. Delivery may be carried out by vehicles that are compatible with the customer's request, during a time-slot (day) in which the customer is available. Further temporal constraints may be incurred due to the customer imposing strict deadlines e.g. $i$ must receive his order before day $s$ or according to the availability of the goods (in some cases goods may be available only after day $s'$). The distance $r_{ij}$ is known for each pair of nodes. The average speed is made equal to a constant $v$ known in advance. A set of vehicles $V$ is located at the depots. The capacity $C_v$, the cost per min of usage $\mu_v$, and the depot $d$ where the vehicle is located are known for each vehicle. We define as $\alpha$ the maximum duration of a route. A set of possible routes $K$ is given, where, for each route $k$, the vehicle $v$ to which it is associated is known as is the day $s$ in which it is scheduled. A customer may be assigned to a specific route only if it is followed. A vehicle $v$ is supposed to be located at depot $d$ on day $s$ (for $s > 1$) if $d$ is the arrival depot of the routes followed by $v$ on day $s-1$. If $v$ has not been used on day $s-1$, it is supposed to be located at the arrival depot of its last followed route, or if it has not been used yet, at the depot where it was located at the beginning of the time-horizon. In order to help the reader, the definitions of the variables and constants are summarized hereafter.

### 2.2. Definitions and notations

In order to help the reader, the definitions of variables and constants are summarized hereafter:

| | |
|---|---|
| *Input data:* | |
| $I = \{1 \dots Imax\}$ | Set of customers |
| $D = \{Imax + 1 \dots Dmax\}$ | Set of depots |
| $N = \{1 \dots Imax + Dmax\}$ | Set of nodes ($N = I \cup D$) |
| $K = \{1 \dots Kmax\}$ | Set of routes |
| $V = \{1 \dots Vmax\}$ | Set of vehicles |
| $S = \{1 \dots Smax\}$ | Set of days |
| $M$ | Very large constant |
| $\epsilon$ | Very small constant |
| $\alpha$ | Maximum route duration |
| $v$ | Average speed (expressed in km/h) |
| $q_i$ | Customer's demand $i$ |
| $r_{ij}$ | Distance between node $i$ and node $j$ |
| $v_k$ | Vehicle associated to route $k$ |
| $t_k$ | Day on which route $k$ is scheduled |
| $C_k$ | Capacity of route $k$ (i.e. capacity of vehicle $v$ performing route $k$) |
| $F_{id}$ | Equal to 1 if customer $i$ can be served by depot $d$ |
| $E_{iv}$ | Equal to 1 if customer $i$ can be served by vehicle $v$ |
| $G_{is}$ | Equal to 1 if customer $i$ can be served on day $s$ |
| $l_{kd}$ | Equal to 1 if vehicle $v$ associated to route $k$ is initially located at depot $d$ |
| $\mu_v$ | cost per minute of usage for vehicle $v$ |
| *Variables:* | |
| $X_{ijk}$ | Boolean variable equal to 1 if arc $ij$ is used by route $k$ |
| $Y_{ik}$ | Boolean variable equal to 1 if customer $i$ is served by route $k$ |
| $Z_k$ | Boolean variable equal to 1 if route $k$ belongs to the solution |
| $T_i$ | Time at which node $i$ is visited |
| $L_{kd}$ | Boolean variable equal to 1 if route $k$ can start from depot $d$ |
| $W_k$ | Duration of route $k$ |
| $\Lambda$ | Objective function |

The capacity and characteristics may be different for each vehicle, or, as occurs in many practical cases, vehicles may be grouped into categories (large standard, small standard, refrigerated, freezer, etc.) and it can be assumed that the vehicles belonging to the same category have the same capacity and characteristics. In order to simplify the formulation and to reduce the number of data involved in the problem, the characteristics of the vehicles and customers are not explicitly reported, but a vehicle-customer compatibility matrix is computed in which the customer's needs and vehicle characteristics are matched. A different cost per minute is associated to each category of vehicles, considering that vehicles with specific features (refrigerated or freezer) are characterized by a higher cost per minute than the standard ones, and that larger vehicles have higher costs than smaller ones.

## 3. A Mixed Integer Programming model for MDMPVRPHF

Using the notation introduced above, a MIP model can be formulated to minimize the total cost for delivery operations over a fixed time-horizon as follows:

$$\min \Lambda = \sum_{i \in N} \sum_{j \in N} \sum_{k \in N} \frac{v}{60} r_{ij} \mu_{v_k} X_{ijk} \tag{1}$$

s.t.

$$\sum_{j \in N | j \neq i} X_{ijk} = Y_{ik} \quad \forall i \in I, \ \forall k \in K \tag{2}$$

$$\sum_{j \in N | j \neq i} X_{jik} = Y_{ik} \quad \forall i \in I, \ \forall k \in K \tag{3}$$

$$\sum_{k \in K} Y_{ik} = 1 \quad \forall i \in I \tag{4}$$

$$\sum_{j \in N | j \neq i} X_{djk} \leqslant L_{kd} \quad \forall k \in K, \ \forall d \in D \tag{5}$$

$$\sum_{j \in N | j \neq i} \sum_{d \in D} X_{djk} = Z_k \quad \forall k \in K \tag{6}$$

$$\sum_{j \in N | j \neq i} \sum_{d \in D} X_{jdk} = Z_k \quad \forall k \in K \tag{7}$$

$$\sum_{i \in I} Y_{ik} \leqslant M \cdot Z_k \quad \forall k \in K \tag{8}$$

$$X_{iik} = 0 \quad \forall i \in I \ \forall k \in K \tag{9}$$

$$T_i \geqslant T_j + \frac{v}{60} r_{ij} X_{jik} - M(1 - X_{jik}) \quad \forall i \in I, \ \forall j \in N, \ \forall k \in K \tag{10}$$

$$T_d = 0 \quad \forall d \in D \tag{11}$$

$$\sum_{i \in I} q_i Y_{ik} \leqslant C_k \quad \forall k \in K \tag{12}$$

$$W_k \leqslant \alpha \quad \forall k \in K \tag{13}$$

$$W_k = \sum_{i \in N} \sum_{j \in N} \frac{v}{60} r_{ij} X_{ijk} \quad \forall k \in K \tag{14}$$

$$L_{kd} = l_{kd} \quad \forall k \in K | t_k = 1 \ \forall d \in D \tag{15}$$

$$\sum_{d \in D} L_{kd} \leqslant 1 \quad \forall k \in K \tag{16}$$

$$L_{kd} = l_{kd}(1 - Z_{k-Vmax}) + \sum_{j \in N} \sum_{\substack{w \in K | \\ v_w = v_k \\ t_w = t_k - 1}} X_{jdw} \quad \forall k \in K | t(k) \geqslant 1, \ \forall d \in D \tag{17}$$

$$L_{kd} \leqslant \sum_{j \in N} \sum_{\substack{w \in K | \\ v_w = v_k \\ t_w = t_k - 1}} X_{jdw} + l_{kd} \quad \forall k \in K | t(k) \geqslant 1, \ \forall d \in D \tag{18}$$

$$L_{kd} \geqslant \sum_{j \in N} \sum_{\substack{w \in K | \\ v_w = v_k \\ t_w = t_k - 1}} X_{jdw} \quad \forall k \in K | t(k) \geqslant 1, \ \forall d \in D \tag{19}$$

$$Y_{ik} \leqslant \sum_{d \in D} F_{id} L_{kd} \quad \forall i \in I, \ \forall k \in K \tag{20}$$

$$Y_{ik} \leqslant E_{iv_k} \quad \forall i \in I, \ \forall k \in K \tag{21}$$

$$Y_{ik} \leqslant G_{it_k} \quad \forall i \in I, \ \forall k \in K \tag{22}$$

$$X_{ijk} \in \{0, 1\} \quad \forall i \in N, \ \forall j \in N, \forall k \in K \tag{23}$$

$$Y_{ik} \in \{0, 1\} \quad \forall i \in I, \forall k \in K \tag{24}$$

$$Z_k \in \{0, 1\} \quad \forall k \in K \tag{25}$$

$$L_{kd} \in \{0, 1\} \quad \forall k \in K, \ \forall d \in D \tag{26}$$

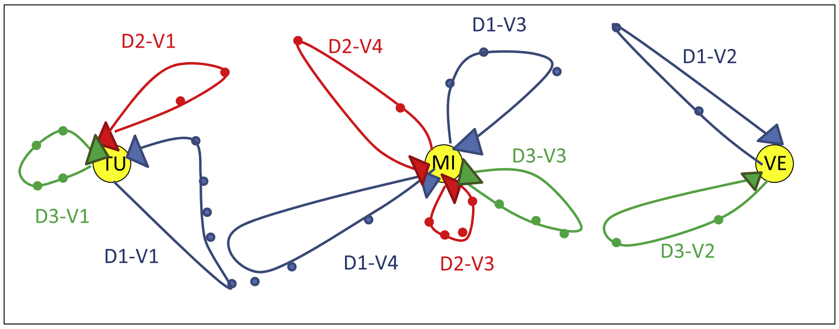

**Fig. 3.** Solution of the Multi-Period Multi-Depot with the constraint of a vehicle having to return to the starting depot.

The objective function that minimizes the total cost is reported in (1). Constraints (2) and (3) ensure that a customer is only visited on a route if he/she has been assigned to that route, while constraint (4) implies that each customer must be assigned to one and only one route. A customer $i$ can only be served by route $k$ if he/she can be served by the depot $d$ from which $k$ starts, as stated in constraint (5). Constraints (6) and (7) imply that if route $k$ is followed, it must start from and end in a depot, and, on the other hand, if it is not followed, the vehicle cannot exit or enter any depot. Constraint (8) states that a customer can be assigned to a route only if that route exists. Subtour elimination is guaranteed by constraints (9)–(11). Limits on the duration of the routes are imposed by (13) and (14), while vehicle capacity satisfaction is ensured by constraint (12). Constraints (15)–(19) allow one to determine the starting depot for each route, which depends on the initial location of the vehicle and the routes it followed on the previous days. Constraints (20)–(22) check customer-route compatibility. Finally, Constraints (23)–(26) specify the variable domain.

## 4. A Large Neighborhood Search based Matheuristic approach

The Large Neighborhood Search heuristic (LNS), belongs to the class of heuristics that is known as Very Large Scale Neighborhood search (VLSN) algorithms, as stated in Ahuja et al. (2002). All VLSN algorithms are based on the observation that searching a large neighborhood results in finding local optima of high quality, and hence a VLSN algorithm may return better solutions. However, searching a large neighborhood is particularly time consuming, hence various filtering techniques are used to limit the search. In VLSN algorithms, the search is usually restricted to a subset of the solutions belonging to the neighborhood that can be searched efficiently; unlike what happens in other VLSN, the neighborhood in LNS is implicitly defined by the moves used to destroy and repair an incumbent solution. For a detailed survey on LNS applications to routing problems the reader may refer to Pisinger and Ropke (2010). The destroy operators may be defined in different ways. For routing problems, for instance, a destroy operator could consist in breaking down $k$ routes and leaving the others unvaried, or in removing a fixed percentage of the arcs in the current solution. A random (or randomized) component is used to select the arcs that have to be removed. The repair method rebuilds a feasible solution, starting from the partially destroyed one. Generally, a greedy construction heuristic is used to rebuild the solution. This is very fast but not always very accurate, since only a sample solution is analyzed in the neighborhood. An enhanced version of LNS, the Adaptive Large Neighborhood Search (ALNS), was proposed in Ropke and Pisinger (2006). The ALNS extends the LNS heuristic by allowing multiple destroy and repair operators to be used within the same search. A destroy operator is chosen at each iteration, on the basis of choosing probabilities. These probabilities are adjusted dynamically during the search process, according to the performance obtained by the operators in the previous iterations.

The innovative aspect of the LNS proposed in this paper concerns the possibility of exploring the whole neighborhood in a reasonably small computational time. In fact, the Large Neighborhood Search is exploited directly by the model. In this way, it is possible to obtain the local minimum with respect to the considered neighborhood, and this makes the intensification phase of the algorithm more powerful and precise. A detailed explanation of how the proposed matheuristic works is given in the following subsection.

## 5. Algorithm description

The proposed matheuristic works on the customers-to-route assignment variables $Y_{ik}$. More precisely, $p$ customers are selected at each iteration of the algorithm; all the other $|I| - p$ customers are assigned to the same route to which they are assigned in the current solution, while the selected $p$ ones are left free to be assigned to any route. In other words, as $P$ is the set of selected customers, and $Y*_ik$ is the value of the assignment variables in the current solution, the following additional constraints are imposed:

$$Y_{ik} = Y^*_{ik} \quad \forall i \in I - P \tag{27}$$

The resulting problem is an ultra-constrained version of the original problem which, for the proper value of $p$, can be solved to optimality in a few seconds by solving the model with any commercial solver. The parameter $p$ can be arbitrary chosen, bearing in mind that small values of $p$ lead to a limited perturbation, with the risk of remaining trapped in local minima, while very large values generate a Large Neighborhood Search which cannot be easily and exhaustively explored in a short computational time. The way by which the $p$ customers that have to be removed are selected is given by an operator. Different destroy operators may be defined. For instance, a *random removal* operator removes $p$ customers randomly, while a *cluster removal* one removes customers located in the same zone in the customer area. A local minimum with respect to a neighborhood is not necessarily a local neighborhood with respect to another neighborhood. For this reason, combining different neighborhood strategies could help to overcome local minima and to obtain high quality solutions. A destroy operator is chosen at each iteration, according to the given probabilities, and the best repaired solution in the neighborhood is found by solving the resulting ultra-constrained problem to optimality. Probabilities are updated during the search process, taking into account information on the operator's performance in the previous iterations. In more detail, each time a destroy operator, $\omega_*$, fails to improve the current best solution, its probability of being chosen, $\Pi_{\omega_*}$ is decreased according to the following rule:

$$\Pi_{\omega_*} = \Pi_{\omega_*} - \delta \tag{28}$$

$$\Pi_{\omega} = \Pi_{\omega} + \frac{\delta}{N_\Omega} \quad \forall a \in \Omega - \omega_* \tag{29}$$

$\Omega$ being the set of defined destroy operators. The larger the value of $\delta$, the greater the penalty inflicted on the non-successful operator. The newly obtained solution, which is the local minimum in the neighborhood, is taken as the current solution for the next iteration. The procedure terminates after a maximum number of iterations, *MAXITER*, or after a maximum number of iterations without any improvement, *MAXNOIMPROVE*, which are given parameters of the algorithm. This procedure needs an initial feasible solution from which to start, which could be computed in different ways; for instance, with a simple greedy constructive heuristic, with a more complex meta-heuristic or directly by means of the model. The quality of the initial solution is not a crucial issue, because, due to the strong diversification inserted into the algorithm, the matheuristic is capable of exploring regions that are potentially distant, in the solutions space, from the starting point, and of converging to a very good solution, even when starting from a poor quality one. This is actually a strong point of the method. In the present case, it was decided to take the best feasible solution obtained by the model within a given short time limit, TIMELIMIT as the initial solution. The value of this parameter should be short but, at the same time must allow one to find at least one feasible solution. The pseudocode of the procedure is reported in the following Algorithm 1.

**Algorithm 1.** An ALNS for the MDMPVRPHF.

```
run the model with a time limit equal to TIMELIMIT and take the best found solution $S^0$ as the initial solution
set the current solution $S$ equal to $S^0$
**repeat**
  select a destroy operator $\omega_*$
  select a subset $P$ of the customers using $\omega_*$
  **for all** $i \in I - P$ **do**
    assign customer $i$ to route $k$ to which it is assigned in $S$ adding constraints $(Y_{ik} = Y^S_{ik})$
  **end for**
  run the model again until the optimal solution is reached
  **if** the newly obtained solution $S'$ is better than $S$ **then**
    set the current solution $S$ equal to $S^0$
  **else**
    update probabilities
  **end if**
**until** maximum number of iterations, *MAXITER*, is reached or no improvement has been obtained for *MAXNOIMPROVE*
  iterations
```

## 5.1. Destroy operators description

Four destroy operators are described in this section. These operators are used to select the customers that are left free to be assigned to any route, while the other customer assignments are fixed. It should be noted that, even though the customers are forced to be assigned to a route, they are inserted into the current solution; their position and sequence within the route may change freely.

- *Random Removal (RR)*
  The Random Removal operator simply selects $p$ customers at random.
- *Cluster Removal (CR)*
  The Cluster Removal operator has the purpose of selecting a cluster of customers that has to be removed, because exchanging customers located very close to each other would offer a better chance of obtaining improved solutions. The operator randomly selects a customer $p^*$ and removes him/her and the nearest $p - 1$ customers.
- *Pair Removal (PR)*
  The Pair Removal operator has the purpose of selecting pairs of very close customers in order to favor one-to-one customer assignment exchange between routes. This operator selects $\frac{p}{2}$ customers and for each of the customers, $p^*$, the nearest customers $p'^*$ being chosen from the customers that have not yet been selected.
- *Route Destruction (RD)*
  The Route Destruction operator has the purpose of completely destroying routes. A route $k$ is randomly selected and customers belonging to it are selected until the selected number of customers is equal to $p$. If all the customers belonging to $k$ have already been selected and the number of selected customers $s < p$, another route is selected. The sequence according to which customers in the route are selected is chosen randomly.

RR is the most diversification oriented operator since it potentially leads to changes in the assignment in all the routes. (The selected customers may be widely distributed over the customer area.) CR, which operates on a small portion of the customer area, may offer a better chance of obtaining an improvement at the beginning of the search than RR, but, when the solution quality is already high, there is a greater risk of remaining trapped in a local minimum, since it looks at a restricted part of the solution which could already have been optimized. Pair Removal (PR) explores a large neighborhood composed of several small neighborhoods, one for each pair. In this way, it combines diversification with intensification strategies. Finally, RD, which completely removes some routes from the solution, allows one to easily obtain solutions with a lower number of routes, if it is possible to relocate all the customers belonging to a given a route to other routes, with a lower cost respect to the current solution. This operator is more successful in reducing the number of routes than the other operators, and in finding solutions that may rarely be found when using another operator, but, on the other hand, when this reduction is not possible, it has a greater chance of remaining trapped in a local minimum.

The initial probabilities are uniformly set as:

$$\Pi_{\omega_1} = \Pi_{\omega_2} = \Pi_{\omega_3} = \Pi_{\omega_4} = 0.25 \tag{30}$$

The probability updating parameter, $\delta$, has been set equal to 0.01.

## 6. Computational results

In this section we report computational results obtained on instances with different characteristics. All the values of the parameters used in the instances are reported in the following:

Number of customers: 30
Number of depots: 3
Number of vehicles: 6
Number of days: 5
Number of potential routes: number of vehicles $*$ number of days = 30
Maximum route duration ($\alpha$): 660 (expressed in minutes)
Average speed ($v$): 80 (expressed in km/h)

Three different levels of customers-vehicles compatibility and of customer availability have been defined. One instance for each combination of these two parameters have been generated.

The customers-vehicles compatibility levels are:

- High compatibility: around 90% of customer-vehicle assignments are feasible.
- Medium compatibility: around 80% of customer-vehicle assignments are feasible.
- Low compatibility: around 60% of customer-vehicle assignments are feasible.

while the customers availability levels are:

- High availability: around 95% of customer-day assignments are feasible.
- Medium availability: around 70% of customer-day assignments are feasible.
- Low availability: around 30% of customer-day assignments are feasible.

These levels have been constructed in order to represent realistic cases which may arise in practical applications. In fact, availability and compatibility level may sensibly very depending on the analyzed context. For instance, in food delivery to the supermarkets, vehicles compatibility is generally low; in fact, frozen or perishable items require specific vehicles, like freezer or refrigerated ones, which generally are a very small subset of the global fleet. Customers availability, instead, is very high because supermarket do not have strictly deadlines, and may be served each day of the week. In other contexts, like electronic products home delivery (carried out by amazon or other online shopping website), strictly delivery deadline are imposed, but vehicles compatibility is around 100%, because products do not have any specific requirement. Since the goal of this project was to create a tool which can be applied on several contexts, we decided to test the model on instances having different characteristics, in order to recreate realistic situations. These instances are available from the author upon requests (simona.mancini@polito.it).

The following values for the parameters of the algorithm has been used:

Perturbation size $p$: 10
Initial solution time limit *TIMELIMIT*: 10 s
Maximum number of iterations *MAXITER*: 100
Maximum number of iterations without improving *MAXNOIMPROVE*: 10

Computational test have been carried out on a coreI7 processor at 1.8 GHz with 8 Giga of Ram. Matheuristic results are compared with the results obtained running the model on XPRESS 7.3. with a time limit of 3600 s. Detailed results obtained are reported in Table 1, which is organized as follows. First column contains the name of the instance, while customer-vehicle compatibility and customers availability are reported in columns 2 and 3, respectively. In columns 4 and 5 we show the best solution (UB) and the lwer bound (LB) obtained by the model within the time limit. Column 6 reports the percentage gap between UB and LB. As shown in Table 1, the gaps obtained by the model are very large, therefore an improving heuristic is required.

The performance of the matheuristic has been tested in the following way. First, each operator has been tested separately, i.e. the probability of choosing other operators is kept equal to zero. In this way it is possible to evaluate the behavior of the single operator. After that, a particular version, named LNS, of the above presented ALNS, has been tested. In this version operators choosing probabilities, assigned following a uniform distribution, are kept constant during the search process. The scope of this experiment is to analyze how using different neighborhoods would help to escape local minima, keeping in mind that a local minimum respect to a neighborhood may be not a local minimum respect to another neighborhood. Finally, the complete ALNS method has been tested. Since the algorithm presents a random component, each method has been run 10 times on each instance. In Table 2 the results obtained by both the model and the matheuristic are reported. For each version of the matheuristic the value both the best solution obtained over 10 runs and the average one are reported. The last three rows of the table reports averaged values over the 9 instances, the gap with the solution obtained by the model and the gap with the initial solution, respectively. Averaged computational times are around 200 s for all the matheuristic versions.

As shown in Table 2, even applying a single operator results obtained a remarkable improvement, (from 7.64% to 9.31%) than those provided by the model. Moreover, the improvement with respect to the initial solution is above 27% on average, which means that the algorithm is capable to converge to a good solution even starting from a poor one, which is actually a good point. The best overall results are obtained by the ALNS which slightly outperform the LNS. Averaged results over 10 runs are quite near to best results for all methods, showing a general good robustness of the proposed approach, but the robustness of ALNS is sensibly higher than the others. This effect is given by the adaptive part of the method, which let less power to the random component.

## 6.1. Parameters tuning

Further tests have been carried out to tune algorithm parameters like the *TIMELIMIT* for the initial solution, and the number of customers to be removed from the solution at each iteration, $p$. For what concerns the TIMELIMIT, the following values

**Table 1**
MIP model upper and lower bounds obtained after 3600 s.

| Istanza | V-COMP | DAY-AV | Model-LB | Model-UB | Gap (%) |
|---|---|---|---|---|---|
| I1 | High | High | 1292.79 | 1902.85 | 47.19 |
| I2 | High | Medium | 1346.74 | 1848.79 | 37.28 |
| I3 | High | Low | 1359 | 1715 | 26.20 |
| I4 | Medium | High | 1341.11 | 1913.14 | 42.65 |
| I5 | Medium | Medium | 1364.49 | 1779.92 | 30.45 |
| I6 | Medium | Low | 1384.01 | 1673.03 | 20.88 |
| I7 | Low | High | 1422.71 | 1743.33 | 22.54 |
| I8 | Low | Medium | 1650.32 | 1811.37 | 9.76 |
| I9 | Low | Low | 1700.01 | 1796.14 | 5.65 |

**Table 2**
Matheuristic results.

| | Model | IS | RR | | CR | | PR | | RD | | LNS | | ALNS | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Best | Avg | Best | Avg | Best | Avg | Best | Avg | Best | Avg | Best | Avg |
| I1 | 1902.85 | 2290.65 | 1626.89 | 1673.05 | 1655.22 | 1706.59 | 1622.31 | 1736.92 | 1569.43 | 1632.67 | 1551.59 | 1631.92 | 1544.50 | 1600.11 |
| I2 | 1848.79 | 1981.54 | 1544.50 | 1577.43 | 1544.50 | 1554.64 | 1575.22 | 1592.20 | 1544.50 | 1558.17 | 1544.50 | 1550.64 | 1544.50 | 1544.50 |
| I3 | 1715.00 | 2466.64 | 1649.92 | 1713.15 | 1602.29 | 1645.21 | 1649.92 | 1731.14 | 1559.37 | 1620.28 | 1545.63 | 1606.71 | 1544.50 | 1563.97 |
| I4 | 1913.14 | 1975.44 | 1574.42 | 1594.07 | 1574.42 | 1614.33 | 1564.47 | 1595.87 | 1557.79 | 1585.80 | 1557.79 | 1579.15 | 1557.79 | 1569.32 |
| I5 | 1779.92 | 2529.48 | 1720.08 | 1753.41 | 1667.61 | 1737.07 | 1701.99 | 1784.32 | 1645.08 | 1685.69 | 1645.08 | 1712.14 | 1618.47 | 1676.17 |
| I6 | 1673.03 | 2377.26 | 1582.56 | 1612.40 | 1582.56 | 1601.92 | 1582.56 | 1598.65 | 1582.56 | 1643.59 | 1582.56 | 1587.17 | 1582.56 | 1586.05 |
| I7 | 1743.33 | 2408.86 | 1646.54 | 1661.39 | 1638.60 | 1659.79 | 1655.77 | 1733.41 | 1627.52 | 1672.70 | 1627.52 | 1651.92 | 1627.52 | 1648.63 |
| I8 | 1811.37 | 2257.23 | 1809.75 | 1863.60 | 1842.67 | 1851.51 | 1791.66 | 1856.96 | 1778.85 | 1862.80 | 1778.85 | 1849.40 | 1778.85 | 1802.18 |
| I9 | 1796.14 | 2236.9 | 1791.71 | 1848.05 | 1784.62 | 1826.98 | 1872.46 | 1900.76 | 1812.05 | 1890.53 | 1784.62 | 1827.02 | 1778.85 | 1805.09 |
| Avg | 1798.17 | 2280.44 | 1660.71 | 1699.62 | 1654.72 | 1688.67 | 1668.48 | 1725.58 | 1630.79 | 1683.58 | 1624.24 | 1666.23 | 1619.73 | 1644.00 |
| Gap model | | | 7.64% | 5.48% | 7.98% | 6.09% | 7.21% | 4.04% | 9.31% | 6.37% | 9.67% | 7.34% | 9.92% | 8.57% |
| Gap IS | | | 27.18% | 25.47% | 27.44% | 25.95% | 26.84% | 24.33% | 28.49% | 26.17% | 28.78% | 26.93% | 28.97% | 27.91% |

**Table 3**
ANLS results for different values of TIMELIMIT.

| Instance | Initial solution | | | ALNS | | |
|---|---|---|---|---|---|---|
| | $t = 10$ | $t = 30$ | $t = 60$ | $t = 10$ | $t = 30$ | $t = 60$ |
| I1 | 2290.65 | 2290.65 | 2290.65 | 1544.50 | 1544.50 | 1544.50 |
| I2 | 1981.54 | 1981.54 | 1981.54 | 1544.50 | 1544.50 | 1544.50 |
| I3 | 2466.64 | 2466.64 | 2466.64 | 1544.50 | 1544.50 | 1544.50 |
| I4 | 1975.44 | 1837.27 | 1837.27 | 1557.79 | 1557.79 | 1557.79 |
| I5 | 2529.48 | 1861.88 | 1861.88 | 1618.47 | 1618.47 | 1618.47 |
| I6 | 2377.26 | 1863.4 | 1863.4 | 1582.56 | 1579.33 | 1579.33 |
| I7 | 2408.86 | 2072.14 | 2072.14 | 1627.52 | 1627.52 | 1627.52 |
| I8 | 2257.23 | 2062.63 | 1893.41 | 1778.85 | 1778.85 | 1778.85 |
| I9 | 2236.9 | 2001.82 | 1926.14 | 1778.85 | 1778.85 | 1778.85 |
| Avg | 2280.444 | 2048.663 | 2021.452 | 1619.727 | 1619.368 | 1619.368 |

**Table 4**
ANLS results for different values of *p*.

| Instance | $p = 5$ | | $p = 10$ | | $p = 15$ | |
|---|---|---|---|---|---|---|
| | Best | Avg | Best | Avg | Best | Avg |
| I1 | 1545.95 | 1622.42 | 1544.50 | 1600.11 | 1584.48 | 1664.89 |
| I2 | 1544.5 | 1552.63 | 1544.50 | 1544.50 | 1544.50 | 1570.99 |
| I3 | 1551.59 | 1566.69 | 1544.50 | 1563.97 | 1544.82 | 1562.70 |
| I4 | 1557.79 | 1599.13 | 1557.79 | 1569.32 | 1557.79 | 1601.82 |
| I5 | 1618.47 | 1685.13 | 1618.47 | 1676.17 | 1618.47 | 1690.20 |
| I6 | 1582.56 | 1587.80 | 1582.56 | 1586.05 | 1582.56 | 1594.78 |
| I7 | 1638.77 | 1681.50 | 1627.52 | 1648.63 | 1631.67 | 1651.11 |
| I8 | 1796.94 | 1830.40 | 1778.85 | 1802.18 | 1778.85 | 1834.92 |
| I9 | 1798.93 | 1892.78 | 1778.85 | 1805.09 | 1778.85 | 1833.46 |
| Avg | 1626.167 | 1668.719 | 1619.727 | 1644.003 | 1624.666 | 1667.208 |

**Table 5**
Model results on large size instances.

| Customers | V-Comp | Day-AV | Model LB | Model UB | Gap (%) |
|---|---|---|---|---|---|
| 50 | High | High | 619.46 | 897.15 | 44.83 |
| 50 | High | Medium | 664.68 | 876.38 | 31.85 |
| 50 | High | Low | **1667.84** | **1667.84** | 0.00 |
| 75 | High | High | 825.42 | 2584.52 | 213.12 |
| 75 | High | Medium | 867.2 | 2584.52 | 198.03 |
| 75 | High | Low | 1929.56 | 2337.66 | 21.15 |

**Table 6**
ALNS results on large size instances.

| Customers | V-Comp | Day-AV | Model | Init sol | ALNS |
|---|---|---|---|---|---|
| 50 | High | High | 897.15 | 4066.3 | 868.13 |
| 50 | High | Medium | 876.38 | 3078.19 | 867.1 |
| 50 | High | Low | **1667.84** | 2784.41 | **1667.84** |
| 75 | High | High | 2584.52 | 3048.03 | 1396.01 |
| 75 | High | Medium | 2584.52 | 3048.03 | 1396.01 |
| 75 | High | Low | 2337.66 | 3048.03 | 2290.63 |
| Avg | | | 1824.68 | 3178.83 | 1414.29 |

have been tested: 10, 30 and 60 s. The goal of this experiment is to analyze whether the initial solution quality influence the final results of the ALNS. As shown in Table 3, giving a larger time limit, the quality of the initial solution sensibly improve (more than 20%), but the final solution found by the ALNS is practically the same, and therefore, the most appropriate choice is to use a very short time limit for the initial solution search phase. This property is an important strength point of the algorithm. In fact, the ALNS is able to reach the same high quality solution even starting from a poor one, that is a proof of the robustness of the algorithm.

Another interesting test has been carried out, regarding the perturbation size, *p*. Three different values of *p* have been tested, $p = 5$, 10, 15. As shown in Table 4, results obtained with $p = 10$ are slightly better than the ones obtained with the other two values. Furthermore, this results are obtained in computational times which are up to four times lower. In fact, with $p = 5$ the neighborhood generated is smaller, therefore the convergence to the best solution is slower (i.e. much more iterations of the algorithm are necessary), while with $p = 15$ we obtain very large neighborhoods which may require long computational times to be exhaustively explored, therefore a time limit must be imposed for the exploration of each neighborhood, and in this way, it would be possible to miss some interesting solutions during the search phase. Basing on these experiments, the choice of $p = 10$ is the most appropriate.

### *6.2. Large size instances*

Further computational tests have been carried out on larger size instances, with 50 and 75 customers. The instances are characterized by high vehicle compatibility while customer availability varies (high–medium–low). Instances with this characteristics are the most difficult to be solved by the model and the most challenging for the ALNS. Lower and upper bound obtained by the model within a time limit of 3600 s are reported in Table 5. Optimal solutions values are marked in bold.

Results reported in Table 6 ALNS show the high quality performances of ALNS, that (within an averaged computational time of 400 s), obtain an averaged percentage improvement of 55% respect to the initial solution and strongly outperforms the model (22% of improvement on average). Optimal solutions values are marked in bold.

## 7. Conclusions and future developments

The Multi Depot Multi Period Vehicle Routing Problem with Heterogeneous Fleet (MDMPVRPHF) has been introduced and formalized in this paper. This is a new Vehicle Routing Problem that arise in real-life contexts, in which the goal is to carry out delivery operations at the minimum cost, while respecting constraints due to driver scheduling, customer/vehicle compatibilities and customer availability. A heterogeneous fleet, composed of vehicles characterized by different capacity, properties and usage costs, has been considered. A MIP formulation and an Adaptive Large Neighborhood Search based Matheuristic have been proposed and computational tests have been carried out on instances with different properties representing realistic distribution contexts. The results obtained from computational tests show the efficacy and the effectiveness of the proposed approach. The most innovative aspect of this approach is the capability of exhaustively exploring a large (or very-large) neighborhood in a very short computational time, which make the LNS framework much more effective than the traditional implementation of LNS in which only a subset of the solutions in the considered neighborhood are analyzed. Another good point of this approach is that it is extremely portable, i.e. it can easily be adapted to other Vehicle Routing Problems dealing with assignment variables, such as 2E-VRP, in which customers are assigned to intermediate facilities, or standard Multi-Depot and Location-Routing problems.

Future developments in this field could address the introduction of customer product compatibilities in order to make the problem even more realistic. This could help to obtain better representative cases in which products which cannot travel in the same vehicle may be requested by the customers, i.e. toxic materials and food. Last but not least, packing constraints could be considered if items of different shapes and dimensions are available.

## References

Ahuja, R., Ergun, ., Orlin, J., Punnen, A., 2002. A survey of very largescale neighborhood search techniques. Discrete Appl. Math. 123, 75–102.
Archetti, C., Speranza, M., 2008. The vehicle routing problem: latest advances and new challenges. In: The Split Delivery Vehicle Routing Problem: A Survey. Springer, Berlin, pp. 103–122, Chapter.
Baldacci, R., Battarra, M., Vigo, D., 2008. The vehicle routing problem: latest advances and new challenges. In: Routing a Heterogeneous Fleet of Vehicles. Springer, Berlin, pp. 3–27, Chapter.
Brandao, J., 2011. A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem. Comput. Oper. Res. 38, 140–151.
Caric, T., Galic, A., Fosin, J., Gold, H., Reinholz, A., 2008. Vehicle routing problem. In: A Modelling and Optimization Framework for Real-World Vehicle Routing Problems, I-Tech, Vienna, Austria, pp. 15–34 (Chapter).
Choi, E., Tcha, D., 2007. A column generation approach to the heterogeneous fleet vehicle routing problem. Comput. Oper. Res. 34, 2080–2095.
Confessore, G., Galiano, G., Stecca, G., 2008. Manufacturing systems and technologies for the new Frontier. In: An Evolutionary Algorithm for Vehicle Routing Problem with Real Life Constraints. Springer, Berlin, pp. 225–228, Chapter.
Crainic, T., Ricciardi, N., Storchi, G., 2009. Models for evaluating and planning city logistics systems. Transport. Sci. 43, 432–454.
Crainic, T., Perboli, G., Mancini, S., Tadei, R., 2010. Two-echelon vehicle routing problem: a satellite location analysis. Proc.: Soc. Behav. Sci. 2, 5944–5955.
Crainic, T., Perboli, G., Mancini, S., Tadei, R., 2012. Impact of generalized travel costs on satellite location in two-echelon vehicle routing problem. Proc.: Soc. Behav. Sci. 39, 195–204.
Francis, P., Smilowitz, K., Tzur, M., 2008. The vehicle routing problem: latest advances and new challenges. In: The Period Vehicle Routing Problem and its Extensions. Springer, Berlin, pp. 73–102, Chapter.
Goel, A., Gruhn, V., 2008. A general vehicle routing problem. Euro. J. Oper. Res. 191, 650–660.
Gonzalez-Feliu, J., 2012. Cost optimisation in freight distribution with cross-docking: N-echelon location routing problem. Promet – Traffic 24 (2), 143–149.
Hasle, G., Kloster, O., 2007. Geometric modelling, numerical simulation, and optimization. In: Industrial Vehicle Routing. Springer, Berlin, pp. 397–435, Chapter.
Larsen, A., 2001. The Dynamic Vehicle Routing Problem, Ph.D. thesis, DTU Lingby, Denmark.
Mancini, S., 2012. The two-echelon vehicle routing problem. 4OR 10 (4), 391–392.
Perboli, G., Tadei, R., Vigo, D., 2011. The two-echelon capacitated vehicle routing problem: models and math-based heuristics. Transport. Sci. 45, 364–380.
Pillac, V., Gendreau, M., Gueret, C., Medaglia, A., 2013. A review of dynamic vehicle routing problems. Euro. J. Oper. Res. 225 (1), 1–11.
Pisinger, D., Ropke, S., 2010. Handbook of metaheuristics: international series in operations research & management science. In: Large Neighborhood Search. Springer, Berlin, pp. 399–419, Chapter.

Ropke, S., Pisinger, D., 2006. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transport. Sci. 40 (4), 455–472.

Salhi, S., Imran, A., Wassan, N., 2014. The multi-depot vehicle routing problem with heterogeneous vehicle fleet: formulation and a variable neighborhood search implementation. Comput. Oper. Res. 52(B), 314–325.

Semet, F., Taillard, E., 1993. Solving real-life vehicle routing problems efficiently using tabu search. Ann. Oper. Res. 41 (4), 469–488.

Toth, P., Vigo, D., 2002. The Vehicle Routing Problem. S.I.A.M., Philadelphia, PA.

Weise, T., Podlich, A., Gorldt, C., 2009. Natural intelligence for scheduling, planning and packing problems. In: Solving Real-World Vehicle Routing Problems with Evolutionary Algorithms. Springer, Berlin, pp. 29–53, Chapter.

Wen, M., Cordeau, J.F., Laporte, G., Larsen, J., 2010. The dynamic multi-period vehicle routing problem. Comput. Oper. Res. 37 (9), 1615–1623.