Computers & Operations Research 140 (2022) 105644

Contents lists available at ScienceDirect

Computers and Operations Research

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

The Consistent Vehicle Routing Problem with heterogeneous fleet

F. Stavropoulou

School of Business and Management, Liverpool John Moores University, L3 5UG, Liverpool, UK

ARTICLE INFO

Keywords:
Metaheuristics
Heterogeneous fleet
Consistency
Tabu search
Variable neighbourhood descent

ABSTRACT

Customer relationship management is of high importance as it strengthens customer satisfaction. Providing consistent customer service cultivates customer retention and brand loyalty. This paper examines a new customer-oriented routing problem, the Consistent Vehicle Routing Problem with heterogeneous fleet. The objective is to create cost-efficient routing plans, utilising a fixed fleet of vehicles with heterogeneous operational characteristics, variable and fixed costs, while providing consistent customer service over multiple periods. Service consistency consists in person and visiting time consistency. A mathematical model capturing all the attributes of the problem is developed and utilised to solve small-scale instances to optimality. To address larger instances, a hierarchical Tabu Search framework is proposed. The proposed metaheuristic utilises an upper-level Tabu Search and an underlying Variable Neighbourhood Descent algorithm. Computational experiments conducted on existing and new benchmark instances show the flexibility, effectiveness and efficiency of the proposed framework. Various managerial insights are derived by examining the cost of imposing customer service consistency as well as customer–vehicle incompatibility constraints.

1. Introduction

Services offered by numerous companies at the customer's location require that service providers, such as drivers, sales representatives, technicians and medical personnel, visit customers on a regular basis. Customer retention is one of the most important challenges organisations are facing, as customer acquisition is around five to six times more expensive, and long-term customers tend to generate higher profits. Providing consistent service with respect to the visiting service provider and the time of visit is a highly desirable feature, leading to increased customer satisfaction and customer retention as it allows the company's employees to build up a rapport with the customers. In addition, consistent service has been shown to increase operational efficiency by enhancing the service providers' familiarity with the customers' environment and decreasing service times. As a result, in recent years, there has been a heightened interest in providing consistent customer-oriented services as it allows companies to stand out from their competitors.

In recent years, customer-oriented Vehicle Routing Problems (VRPs) with consistency features have received significant attention due to their practical importance. Three types of consistency are found in the literature; (i) visiting time consistency, (ii) person consistency and (iii) quantity consistency. VRPs with consistency attributes have been used to model various real-life applications, such as parcel delivery (Groër et al., 2009), home healthcare (Rusell et al., 2011), the transportation of disabled (Feillet et al., 2014) and elderly people (Braekers and Kovacs, 2016), pharmaceutical distribution (Campelo et al., 2019), home meal delivery (Hewitt et al., 2015), home groceries delivery (Song et al., 2020), retail distribution (Ulmer et al., 2020), soft drinks distribution (Rodríguez-Martín et al., 2019), aircraft fleet scheduling (Ioachim et al., 1999) as well as cleaning services (Tarantilis et al., 2012). The existing variants of VRPs with consistency considerations make the assumption that an unlimited number of identical vehicles is available at a central depot. However, this is not a realistic assumption as in most real-life cases a vehicle fleet is likely to be heterogeneous and of limited/fixed size (Hoff et al., 2015; Koç et al., 2016).

In practice, vehicle fleets are rarely homogeneous, with vehicles having different operating, service and depreciation costs, as organisations procure their fleet over a long period of time. Additionally, due to technology advancements, vehicles tend to possess different features and functionality. In recent years, the development of alternative fuel and technology vehicles, such as hybrid or fully electric vehicles, has led to diverse fleet compositions of battery-powered and conventional vehicles. There are also restrictions imposed by the transportation network, customers or environmental regulations. For example, in urban areas, city centres or old villages, narrow streets make the use of smaller and more flexible vehicles necessary. Moreover, there may be cases where servicing a customer requires a vehicle with special equipment (e.g. for loading and unloading). Furthermore, there are special zones in urban areas, such as the ultra low emission zone in

E-mail address: f.stavropoulou@ljmu.ac.uk.

https://doi.org/10.1016/j.cor.2021.105644
Received 28 January 2021; Received in revised form 12 November 2021; Accepted 14 November 2021
Available online 5 December 2021

central London, posing restrictions on the vehicles' noise, gas and particle emissions (Hoff et al., 2015; Vidal et al., 2020; Zbib and Laporte, 2020). All the above highlight that in reality, organisations utilise a limited/fixed number of vehicles with heterogeneous characteristics, such as capacity, speed and operational cost. Therefore, operations managers have to develop routing plans that make the best use of the available resources, i.e. the vehicle fleet, whilst gaining a competitive advantage for their organisation.

To this end, this paper addresses a new VRP of practical importance, the Consistent Vehicle Routing Problem with heterogeneous fleet (HConVRP). The HConVRP aims to minimise overall transportation costs, utilising a heterogeneous vehicle fleet of fixed size, offering consistent customer service over a planning horizon of multiple periods. In the HConVRP context, consistent customer service consists in imposing time and person consistency constraints at the same time, i.e. for customers receiving service more than once within the given planning horizon, the service is performed by the same service provider at roughly the same time. The proposed problem relates closely with the Consistent VRP (ConVRP) and the Heterogeneous VRP (HVRP).

The contribution of this paper is threefold. First, it introduces and models a new VRP with multiple attributes, the HConVRP. This problem aims to make the best use of the available heterogeneous vehicle fleet, while providing consistent customer service. The objective of the HConVRP is to determine a set of cost-efficient routing plans, minimising the total transportation cost, i.e. the fixed and variable cost. Second, a hierarchical Tabu Search (TS) algorithm is developed, incorporating four user-defined parameters. The proposed metaheuristic adopts a hierarchical bi-level search framework that takes advantage of different search landscapes. At the upper level, the solution space is explored on the basis of the customer assignment to service providers/vehicles, using a TS method, while at the lower level the routing of customers is optimised in terms of travelling distance via a Variable Neighbourhood Descent (VND) method. Third, an extensive set of computational experiments is reported. The proposed metaheuristic's performance is evaluated on existing ConVRP instances and proved competitive compared to the state-of-the-art. After generating new benchmark instances for the HConVRP, small-scale instances are solved to optimality. The proposed algorithm obtained the optimal solutions in most cases (worst case performance is 2.16%). Lastly, various managerial insights are discussed by examining the trade-off between the transportation cost and consistent customer service as well as customer–vehicle incompatibilities.

The remainder of the paper is organised as follows. Section 2 discusses the relevant literature, followed by Section 3 that provides the necessary notation and proposes a formal mathematical formulation for the HConVRP. Section 4 presents the proposed solution method. Computational results are presented, discussed and analysed in Section 5. Finally, conclusions are drawn in Section 6.

2. Related works

2.1. Consistency routing

The concept of providing services in a consistent manner has been recently studied in VRP literature. A seminal work in the field is that of Groër et al. (2009) introducing the ConVRP. The ConVRP extends the traditional vehicle capacity and route duration VRP constraints, imposing that the same driver visits the same customers (driver consistency) at approximately the same time on each day they require service (arrival time consistency) over a planning horizon. A number of heuristics have been proposed for addressing the ConVRP. Groër et al. (2009) develop a Record-to-Record travel algorithm, utilising template routes. A template is a set of predetermined artificial routes, containing only the customers with frequent visiting requirements, i.e. requiring to be visited more than once within the planning period, that are used as a guide to design the actual daily schedules. From the algorithmic viewpoint, all the existing heuristics explicitly designed for the ConVRP adopt the template route rationale. Tarantilis et al. (2012) propose a two-level master–slave TS algorithm, in which template routes are improved by TS at the master level, and then the actual daily schedules are constructed and optimised by the slave TS. A fast template-based Adaptive Large Neighbourhood Search algorithm (ALNS) is developed by Kovacs et al. (2014b). ALNS is used to optimise the template routes, and then the actual routes are improved using a truncated 2-opt operator. Xu and Cai (2018) present a two-stage template-based Variable Neighbourhood Search solution framework (VNS), where template routes, including all customers, are improved by VNS and an actual solution is produced if certain cost criteria hold. Recently, a Cluster Column Generation method (CCG) has been developed by Goeke et al. (2019), solving instances of up to 30 customers optimally. The CCG method incorporates a Large Neighbourhood Search (LNS-SP) to obtain upper bounds, which can also be used as a stand-alone solution method for the ConVRP.

In the ConVRP literature, various types of constraints have been adopted. Harder types dictating both driver/carrier and visiting time consistency (Groër et al., 2009; Stavropoulou et al., 2019; Zhen et al., 2020; Mancini et al., 2021), as well as relaxed versions taking into account time consistency only (Feillet et al., 2014) or versions aiming to limit the number of drivers that visit a customer (Kovacs et al., 2015a; Luo et al., 2015; Braekers and Kovacs, 2016). Additionally, in some ConVRP variants consistency is addressed in the objective function, instead of the problem constraints, either in an aggregated form (Sungur et al., 2010; Kovacs et al., 2015a; Ulmer et al., 2020) or as multiple objective functions (Smilowitz et al., 2013; Kovacs et al., 2015b; Lian et al., 2016). Furthermore, Subramanyam and Gounaris (2016) present the Consistent Traveling Salesman Problem (ConTSP), taking into account time consistency constraints only, as a single vehicle is utilised. It is noteworthy that relaxed versions allowing vehicle waiting either at the depot or at customer locations are examined by Kovacs et al. (2014b) and Goeke et al. (2019) and Subramanyam and Gounaris (2017) for the ConVRP and the ConTSP, respectively. This paper adopts the hard types of consistency constraints, dictating both driver and arrival time consistency, whilst vehicles are not allowed to idle. A detailed survey on ConVRPs is presented by Kovacs et al. (2014a).

2.2. Heterogeneous fleet routing

The HVRP is the extension of the VRP where additional decisions on the fleet composition have to be made (Koç et al., 2016). Specifically, the HVRP generalises the traditional VRP taking into consideration a heterogeneous fleet with various vehicle types of different operational characteristics and costs. Two major HVRP variants with different fleet size considerations are the Fleet Size and Mix VRP (FSM) and the Heterogeneous Fixed Fleet VRP (HF). The FSM was introduced by Golden et al. (1984) and assumes an unlimited number of vehicles of each type, whereas Taillard (1999) proposed the HF in which a limited number of vehicles of each type is available. Another classification factor for HVRPs is whether their objective function considers the minimisation of fixed or variable costs or their combination. Thus, combining the two aforementioned criteria, HVRPs can be categorised in five main groups: (a) the FSM with fixed and variable costs (FSM(F,V)), (b) the FSM with fixed costs (FSM(F)), (c) the FSM with variable costs (FSM(V)), (d) the HF with fixed and variable costs (HF(F,V)) and (e) the HF with variable costs (HF(V)). It should be highlighted that the classification and notation proposed by Koç et al. (2016) is followed. This paper belongs to the fourth group, utilising a heterogeneous fleet of fixed size, with the aim to minimise both fixed and variable costs.

HVRPs have been studied to a great extent for more than 30 years and can be used to model a variety of applications mostly occurring in the areas of product delivery and distribution including the distribution of fresh milk either to supermarkets/convenience stores or directly to customers, distribution of ready-made concrete to construction sites, furniture delivery to customers, parcel collection and delivery, waste collection and carton pickup and delivery to customers. It is worthwhile noting that HVRPs can also be used to model problems related to the material flow within construction projects (Koç et al., 2016). A number of papers present variants that involve additional features of practical importance such as time windows (Liu and Shen, 1999; Bräysy et al., 2008; Paraskevopoulos et al., 2008; Koç et al., 2015), multiple depots (Salhi et al., 2014), demand uncertainty (Subramanyam et al., 2020), pickups and deliveries (Qu and Bard, 2014), multiple trips (Wassan et al., 2017), multiple periods (Mancini, 2016), site dependences (Chao et al., 1998), backhauls (Salhi et al., 2013), open routes (Li et al., 2012), green routing (Koç et al., 2014), multiple stacks (Iori and Riera-Ledesma, 2015), multi-compartment vehicles (Zbib and Laporte, 2020) and split deliveries (Ceselli et al., 2019). However, to the best of the author's knowledge, there are no papers combining HVRPs and service consistency. Literature reviews regarding the class of HVRPs can be found in Baldacci et al. (2008), Baldacci et al. (2010), Irnich et al. (2014) and Koç et al. (2016).

3. Problem definition and formulation

3.1. Problem definition

The HConVRP can be defined on a complete undirected graph $G = (N, E)$, where $N = \{0, 1, 2, \dots, n\}$ is the node set and $E = \{(i, j) : i, j \in N, i \neq j\}$ is the edge set. The depot is located at node 0 and the set of customers is denoted by $N_c = N \setminus \{0\}$. A non-negative distance d_{ij} is associated with each edge $(i, j) \in E$, while the corresponding distance matrix $[d_{ij}]$ is symmetric, i.e. $d_{ij} = d_{ji}$, and the triangle inequality is satisfied. A fixed fleet of heterogeneous depot-returning vehicles is available. This vehicle fleet consists of H different vehicle types, with $K = \{1, \dots, H\}$. For each vehicle type $k \in K$, a set M of h_k identical vehicles are available, each having a carrying capacity of Q_k units, a maximum operation duration of T time units and an average speed B_k. As each vehicle type k has a different speed, a travelling time $t^k_{ij} = d_{ij}/B_k$ is defined. Each vehicle type is also associated with a fixed cost F_k, which corresponds to the rental or capital amortisation and insurance cost, and a variable cost V_k, which corresponds to the fuel, driver and maintenance cost. The latter means that for each edge $(i, j) \in E$ there is a corresponding traversal cost $c^k_{ij} = d_{ij} \times V_k$ associated with vehicle type k. Moreover, there may be compatibility restrictions between a vehicle type and an edge, denoted by u^k_{ij}, i.e. $u^k_{ij} = 1$ if vehicle type k can be used on edge $(i, j) \in E$ and $u^k_{ij} = 0$ otherwise. Each vehicle can perform one route per period $p \in P$, where P is the set of periods $P = \{1, \dots, g\}$, starting and ending at the depot, with a maximum duration T and accumulated load Q_k. Additionally, each customer $i \in N_c$ poses specific service requirements w_{ip} for each period $p \in P$, i.e. $w_{ip} = 1$ if customer i needs to be serviced in period p and 0 otherwise. Each customer must be visited only once in the period $p \in P$ they need to be serviced. Furthermore, for each customer $i \in N_c$ there is a known service time s_{ip} and a positive demand q_{ip}. The customer set N_c can be split into two non-overlapping subsets; the set of frequent customers N_f and the set of non-frequent customers N_{nf}. Driver consistency imposes that each frequent customer $i \in N_f$ must be visited by the same vehicle $m \in M$ over all periods $p \in P$, whereas arrival time consistency dictates that the difference between the earliest and latest vehicle arrival times to a frequent customer $i \in N_f$ must be at most L (it should be noted that vehicles are not allowed to idle). For notational convenience, E_p denotes a reduced set of edges, $E_p = \{(i, j) \in E : w_{ip} w_{jp} = 1\}$ and N_p a reduced set of nodes, $N_p = \{i \in N_c : w_{ip} = 1\}$ for each period $p \in P$.

The objective is to determine a routing plan that minimises the total transportation cost, i.e. the fixed and the variable costs of all utilised vehicles.

3.2. Mathematical formulation

In this paper, a five-index formulation is introduced. More specifically, three groups of variables are utilised. Each binary variable x^{km}_{ijp} counts the number of times edge $(i, j) \in E$ is traversed in period p by vehicle m belonging to vehicle type k, binary variables z^m_{ip} indicate if customer i is serviced by vehicle m in period p and continuous variables a^{km}_{ip} depict the arrival time of vehicle m belonging to vehicle type k to customer i in period p in the optimal solution. Given the above representation, the HConVRP can be mathematically depicted as follows:

$$\min \sum_{k\in K}\sum_{m\in M}\sum_{p\in P}\sum_{j\in N_c} F_k x^{km}_{0jp} + \sum_{p\in P}\sum_{(i,j)\in E}\sum_{m\in M}\sum_{k\in K} c^k_{ij} x^{km}_{ijp} \tag{1}$$

Subject to

$$\sum_{m\in M} z^m_{ip} = w_{ip} \quad \forall p \in P, i \in N_c \tag{2}$$

$$\sum_{i\in N_p}\sum_{m\in M} x^{km}_{i0p} = \sum_{j\in N_p}\sum_{m\in M} x^{km}_{0jp} \leq h_k \quad \forall k \in K, p \in P \tag{3}$$

$$\sum_{j\in N}\sum_{k\in K} x^{km}_{ijp} = \sum_{j\in N}\sum_{k\in K} x^{km}_{jip} = z^m_{ip} \quad \forall i \in N (i \neq j), p \in P, m \in M \tag{4}$$

$$1 - x^{km}_{ijp} - x^{km}_{jip} \geq z^m_{ip} - z^m_{jp} \quad \forall (i, j) \in N_p \times N_p : i \neq j, k \in K, m \in M, p \in P \tag{5}$$

$$\sum_{i\in N}\sum_{j\in N_c} q_{jp} x^{km}_{ijp} \leq Q_k \quad \forall m \in M, k \in K, p \in P \tag{6}$$

$$z^m_{0p} = 1 \quad \forall m \in M, p \in P \tag{7}$$

$$a^{km}_{0p} = 0 \quad \forall k \in K, m \in M, p \in P \tag{8}$$

$$a^{km}_{ip} + x^{km}_{ijp}(s_{ip} + t^k_{ij}) - (1 - x^{km}_{ijp})T \leq a^{km}_{jp} \quad \forall i \in N, j \in N_c, p \in P, k \in K, m \in M \tag{9}$$

$$a^{km}_{ip} + x^{km}_{ijp}(s_{ip} + t^k_{ij}) + (1 - x^{km}_{ijp})T \geq a^{km}_{jp} \quad \forall i \in N, j \in N_c, p \in P, k \in K, m \in M \tag{10}$$

$$a^{km}_{ip} + w_{ip}(s_{ip} + t^k_{i0}) \leq w_{ip} T \quad \forall i \in N_c, p \in P, m \in M, k \in K \tag{11}$$

$$|a^{km}_{ip} - a^{km}_{ip'}| \leq L \quad \forall i \in N_p \cap N_{p'}, p \in P, p' \in P : p \neq p', k \in K, m \in M \tag{12}$$

$$x^{km}_{ijp} \leq u^k_{ij} \quad \forall (i, j) \in E_p, k \in K, m \in M, p \in P \tag{13}$$

$$z^m_{ip}, x^{km}_{ijp} \in \{0, 1\}, a^{km}_{ip} \geq 0 \quad \forall i, j \in N, k \in K, m \in M, p \in P \tag{14}$$

$$w_{ip} t^k_{0i} \leq a^{km}_{ip} \leq T - s_{ip} - t^k_{i0} \quad \forall i \in N_c, p \in P, m \in M, k \in K \tag{15}$$

The objective function (1) minimises the total transportation cost. Constraints (2) ensure that customers are visited in each period they require service. Constraints (3) impose that no more than h_k vehicles of type $k \in K$ leave and return to the depot in every period. Constraints (4) ensure route connectivity. Driver consistency constraints (5) impose that each frequent customer is serviced by the same vehicle. Constraints (6) ensure the capacity restrictions for each vehicle are respected, while all vehicles must start from the depot, according to constraints (7). The vehicle arrival times to the depot and to all serviced customers are calculated with constraints (8), (9), (10) and (11). They constitute the Miller–Tucker–Zemlin (MTZ) constraints for time duration and, thus, function also as subtour elimination constraints. It should be noted that constraints (10) impose that there is no vehicle idling. If we wanted to allow vehicle waiting at a location to provide arrival time consistency, then constraints (10) should be removed. Arrival time consistency ensures that the difference between the earliest and latest arrival time to a frequent customer is at most L time units. This is modelled using constraints (12). Constraints (13) dictate that only compatible vehicle types traverse the corresponding edges $(i,j) \in E$. Finally, the last sets impose binary conditions to x and z variables as well as lower and upper bounds for the continuous a-variables.

4. Solution method

4.1. Basic concept and solution framework

In broad terms, multi-period VRPs are routing problems in which vehicle routes must be determined over a predefined planning horizon of p periods. A common practice in literature is to decompose a multi-period VRP into $|P|$ separate VRPs and solve them individually. However, contrary to traditional multi-period VRPs, the HConVRP cannot be decomposed into independent VRPs, since the daily routing plans are closely interrelated due to the driver and arrival time consistency constraints. Thus, two types of decisions have to be made for this problem: first the customer-to-vehicle assignment and then the customer sequencing within the different vehicle routes. The customer-to-vehicle assignment decision is central to the HConVRP as this has to remain consistent throughout the planning horizon and subsequently affects all the sequencing decisions, as well. Given that in the HConVRP the available vehicles have different operational characteristics and costs, the customer-to-vehicle assignment decisions become more complex. To this end, two search landscapes are explored on the basis of the customer-to-vehicle assignment and the customer sequencing decision sets. In order to provide a good distribution of the search effort devoted to the aforementioned search landscapes, a hierarchical solution framework with two levels was developed. On the upper level, the customer-to-vehicle assignment decisions are first made and, given these, then the lower level deals exclusively with the sequencing counterpart of the problem.

From the algorithmic point of view, the proposed hierarchical TS (HTS) can be seen as a multi-start trajectory local search approach, utilising a short-term memory and multiple neighbourhood structures to allow a more thorough exploration of the search space. An issue encountered by local search methods is that they tend to limit themselves to a portion of the search space. To overcome this problem, the proposed HTS, apart from utilising a short-term memory, incorporates a multi-start mechanism and exploits a variety of neighbourhood structures to diversify the search process. A greedy randomised constructive heuristic is employed to generate an initial solution, which is further improved by the HTS algorithm. In an effort to deal with and achieve the desirable consistency in the routing plans the concept of template routes was adopted to construct cost-efficient initial solutions quickly. As reported in Stavropoulou et al. (2019), it is meaningful to employ the template routes as an algorithmic component in frameworks addressing VRPs with consistency considerations as frequent customer pairs tend to appear in the same order in all common periods. For this reason, in this paper, template routes, containing only frequent customers, are used to create initial feasible solutions. A key element of the HTS solution framework is that it operates on hierarchical levels. The HTS algorithm exploits the upper level neighbourhood structures only, whereas, within the HTS algorithm, a VND heuristic is employed to optimise the lower level sequencing counterpart of the HConVRP.

The pseudocode of the proposed HTS metaheuristic is presented in Algorithm 1. After determining the frequent and non-frequent customer sets (Lines 2–3), a greedy randomised constructive heuristic is utilised to generate an initial feasible solution (Line 5). Then, HTS algorithm is triggered to improve the initial solution (Line 6) and the best obtained solution s^* is updated (Lines 7–9). The HTS solution framework terminates after Θ iterations (Line 4) (termination criterion) and the best encountered solution s^* is returned. Input parameters I_{TS} and ω_t control the termination condition for the HTS (number of iterations without observing any improvement) and the tabu tenure (tabu list size), while β and γ control the deterministic neighbourhood oscillation, respectively.

Algorithm 1: HTS solution framework

Input: $I_{TS}, \omega_t, \beta, \gamma$
1 $s, s^* \leftarrow \emptyset$
2 $N_f \leftarrow$ Build set() // Frequent customers
3 $N_{nf} \leftarrow$ Build set() // Non-Frequent customers
4 **for** $i \leftarrow 1$ to Θ **do**
5 $s \leftarrow$ Constructive Heuristic(N_f,N_{nf})
6 $s' \leftarrow$ HTS(s,I_{TS},ω_t,β,γ)
7 **if** $f(s') < f(s^*)$ **then**
8   $s^* \leftarrow s'$
9 **end if**
10 **end for**
Output: s^*

4.2. Constructive heuristic

Initial feasible solutions are generated via a randomised insertion heuristic in two phases. In the first phase, the template routes, including frequent customers only, are constructed, providing the basis for the second phase. In the second phase, for each day, partial vehicle routing schedules are determined by removing the frequent customers that do not require service on that day. Then, the non-frequent customers are routed, using a cheapest insertion criterion, forming the initial routing plans. Algorithm 2 provides an overview of the proposed constructive heuristic. The generated solutions contain a list of daily schedules with the corresponding customers' visiting sequence.

Firstly, the frequent customers are sorted in random order (Line 3). For each frequent customer all feasible template insertion positions are considered and the least-cost one is selected (Lines 4–11). It is highlighted that each position's insertion cost is calculated taking into account the travelling cost for all periods the customer under consideration requires service.

In the next stage, for each period of the planning horizon, partial vehicle routes are created by adopting the template route schedule and removing the frequent customers that do not require service in that period (Lines 12–14). Subsequently, the non-frequent customers requiring service in that period are identified and sorted in random order (Line 15). Adopting the same rationale discussed above, for each non-frequent customer all feasible insertion positions are examined and the customer is inserted in the least cost one (Lines 16–23).

It is worth noting that the constructed template routes can be infeasible in terms of the vehicle capacity and route duration overall as there might be frequent customers that do not require visits on certain days. However, infeasible daily routes are not allowed during the solution construction procedure in terms of vehicle capacity, route duration and service consistency. In the case that customer–vehicle compatibility restrictions apply, customers are only allowed to be inserted in compatible vehicle routes. The constructive heuristic terminates after routing all non-frequent customers. It should be noted that in this paper it is assumed that the available fleet is sufficient to visit all customers.

4.3. Neighbourhood structures

The proposed framework relies on two different groups of neighbourhood structures: the ones affecting the frequent customer-to-vehicle assignment and the ones affecting the total travelling distance. The inter-route neighbourhood structures of the first group, namely ChangeVehicle, SwapVehicle and ChangeVehicleChain,

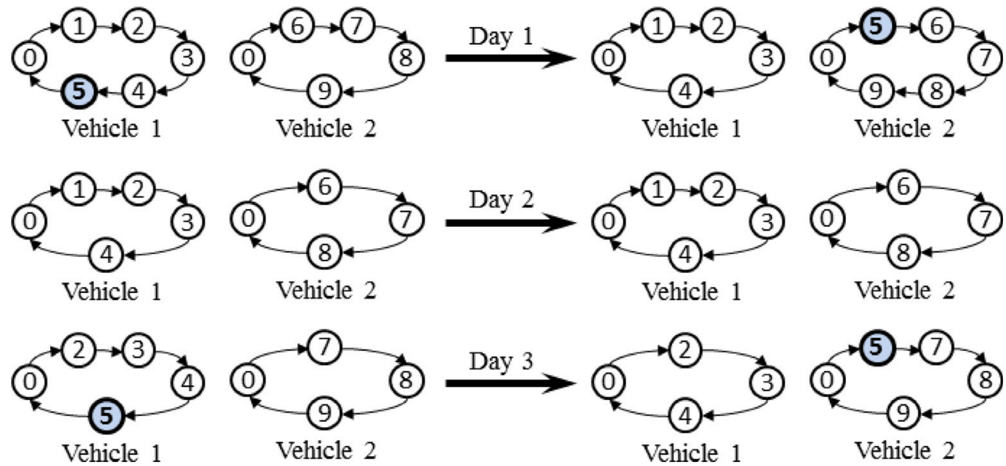

Fig. 1. ChangeVehicle move.

Algorithm 2: Randomised constructive heuristic scheme

```
Input: N_f, N_nf
s ← ∅ //Generate empty solution
TR ← ∅ //Generate an empty template route
  schedule
Randomly Sort Frequent Customers(N_f)
for i ← 1 to |N_f| do          For every Node
  for k ← 1 to H do            For every Vehicle Type
    for m ← 1 to h_k do        For every Vehicle
      v ← Find Feasible Least-Cost Vehicle
    end for
  end for
  TR ← Insert Customer(v) // Route customer to
  vehicle
end for
for p ← 1 to |P| do
  ds_p ← ∅ //Generate an empty daily schedule
  ds_p ← Initialisation(TR,p,N_f) //Adaptation of
  template
  Randomly Sort Non-Frequent Customers of Period p(N^p_nf)
  for i ← 1 to |N^p_nf| do
    for k ← 1 to H do
      for m ← 1 to h_k do
        v ← Find Feasible Least-Cost Vehicle
      end for
    end for
    ds_p ← Insert Customer (v) // Route customer to
  vehicle
  end for
  s ← Add(ds_p)
end for
Output: s
```

change the decisions made regarding the vehicles visiting frequent customers (Figs. 1, 2, 3). The ChangeVehicle neighbourhood proposed by Stavropoulou et al. (2019) is adopted and the corresponding move is presented in Fig. 1. The ChangeVehicle consists of altering the assignment of frequent customer i from vehicle m_a to vehicle m_b. In Fig. 1, the vehicle assignment of customer 5 changes for all days of the planning horizon from Vehicle 1 to Vehicle 2.

In a similar manner, the SwapVehicle neighbourhood consists of interchanging the assignment of frequent customers i and j between vehicles m_a and m_b (Fig. 2). As shown in Fig. 2, frequent customer 5 is serviced by Vehicle 1 and customer 8 is serviced by Vehicle 2, before applying the corresponding SwapVehicle move, while afterwards Vehicle 1 services customer 8 and Vehicle 2 services customer 5, respectively. It is noteworthy that the frequent customers under consideration do not have to have the same visiting frequency. In this example, customer 8 needs to be visited every day of the planning horizon, whereas customer 5 requires to be visited only twice.

Finally, a compound move neighbourhood structure is introduced, the ChangeVehicleChain that reassigns frequent customer i from vehicle m_a to vehicle m_b, while at the same time removing frequent customer j from vehicle m_b and assigning them to vehicle m_c, creating a relocation chain (Fig. 3). In Fig. 3, customer 5 is assigned from Vehicle 1 to Vehicle 2, while simultaneously customer 8 is moved from Vehicle 2 to Vehicle 3. Following the same rationale as in the SwapVehicle neighbourhood, the considered customers do not need to have the same visiting frequencies.

For the routing counterpart of the HConVRP, traditional edge-exchange neighbourhood structures are adopted, namely intra- and inter-route 2-Opt, Shift(1,0) and Swap(1,1) (Penna et al., 2019). It should be highlighted that when the driver consistency constraints are active, the inter-route neighbourhoods apply to non-frequent customers only, as the frequent customers' assignment to vehicles has to remain consistent throughout the planning horizon.

From the implementation point of view, the advantage of the proposed neighbourhood structures is that they are simple, flexible and easy to implement. A lexicographic search is followed for their evaluation, focusing only on feasible neighbours and not considering infeasible ones. Emphasis is given on direct feasibility gains to accelerate the evaluation process. For example, subsets of moves are filtered out in advance on the basis of being infeasible in terms of vehicle capacity.

4.4. Hierarchical Tabu Search

The HTS algorithm is used to further improve the initial solutions generated by the randomised constructive heuristic. The proposed HTS implementation is presented in Algorithm 3. HTS performs search trajectories by moving iteratively from a solution s to the best admissible solution s' of a subset $\Phi_y(s)$ of a given neighbourhood structure y. During the search, solutions are allowed to deteriorate to escape from local optima, while the most recently encountered solutions' characteristics are recorded in a short-term memory, known as tabu list, to avoid cycling. The size ω_t of tabu list is called tabu tenure (Lines 5–12). The tabu status of a neighbouring solution is overridden if certain aspiration criteria hold, i.e. when a new local optimum is obtained (Line 14–15). The overall procedure iterates until a maximum number of iterations I_{TS}, without observing any further improvement, is reached (Line 4) and the best encountered solution s^* is returned.

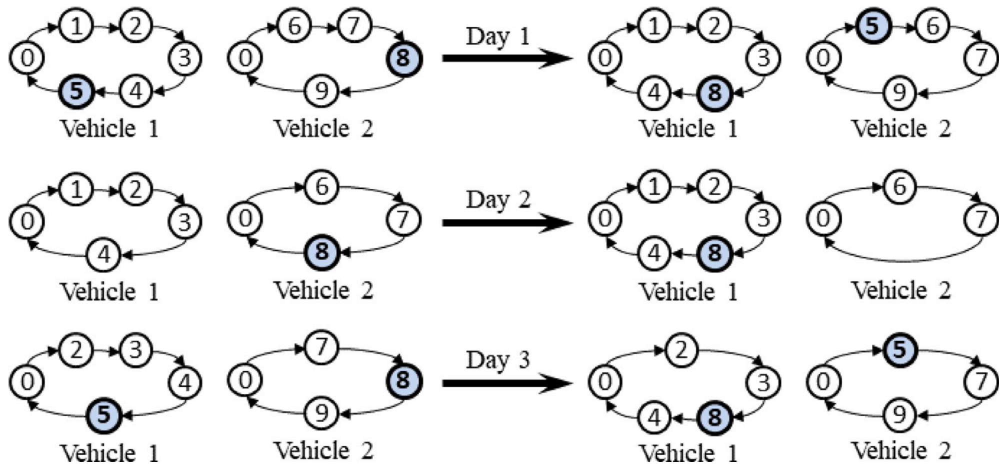

Fig. 2. SwapVehicle move.

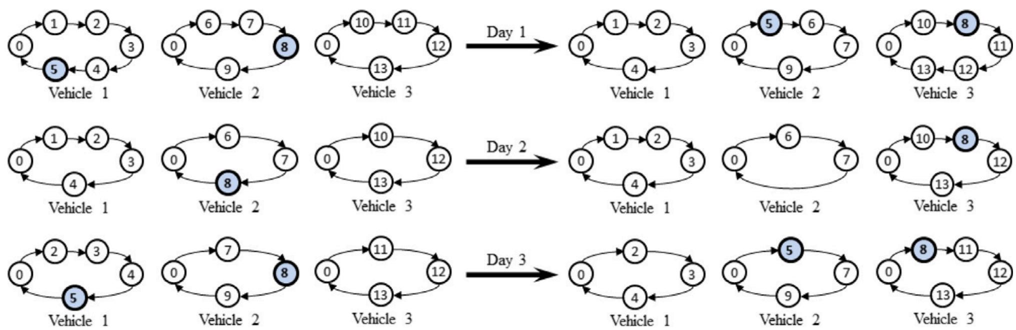

Fig. 3. ChangeVehicleChain move.

Algorithm 3: HTS algorithm

Input: s,I_{TS},ω_t,β,γ
//Set of neighbourhood moves, $y = 1, 2, \dots, y_{max}$//
1 $TM \leftarrow \emptyset$ //`Generate an empty tabu moves set`
2 $s^* \leftarrow s$
3 $s \leftarrow$ VND(s) //`Distance minimisation`
4 **for** $z \leftarrow 1$ to I_{TS} **do**
5 $y \leftarrow 1$
6 **if** z % β ==0 **then**
7 $y \leftarrow 2$
8 **if** z % γ ==0 **then**
9 $y \leftarrow y_{max}$
10 $\Phi_y(s) \leftarrow$ Build Allowed Set(s, y)//`Neighbourhood evaluation`
11 $s \leftarrow \arg\min_{s' \in \Phi_y(s)}\{f(s')\}$
12 $TM \leftarrow$ Update Tabu List(s,y,ω_t)
13 $s \leftarrow$ VND(s) //`Distance minimisation`
14 **if** $f(s) < f(s^*)$ **then**
15 $z \leftarrow 0$, $s^* \leftarrow s$
16 **else**
17 $z \leftarrow z$+1
18 **end if**
19 **end for**
Output: s^*

As far as the neighbourhood oscillation is concerned, a deterministic approach is followed, as neighbourhoods are explored in a predefined order and a best-accept strategy is adopted. The ChangeVehicle neighbourhood is mainly exploited, as it is the quickest to evaluate, and the SwapVehicle and the ChangeVehicleChain neighbourhoods are explored every β and γ iterations, respectively (Lines 5–9), in an effort to diversify local search by exploring larger and more complex neighbourhoods. In this paper, $\beta = 50$ and $\gamma = 75$. If no feasible neighbours exist in the ChangeVehicle neighbourhood exploration then the next available neighbourhood, i.e. the SwapVehicle, is searched and if it is not possible to obtain a feasible neighbour in the SwapVehicle neighbourhood then the ChangeVehicleChain neighbourhood is explored. If there are no feasible neighbours in any of the explored neighbourhood structures then HTS terminates. It is noteworthy that at each HTS iteration the VND algorithm is applied to improve the routing of the customers and to "polish" the current solution in terms of the total travelling distance (Line 13).

4.5. *Variable neighbourhood descent*

The VND algorithm is used to reduce the total travelling cost of a given solution s. In broad terms, this algorithm follows a deterministic search framework that explores a set of ordered neighbourhood structures of increasing cardinality. The rationale of this framework is to apply a search strategy by systematically changing neighbourhood structures. VND seeks to explore the solution space by starting from a predefined neighbourhood y and moving from a solution s to the best improving solution s' of neighbourhood $\Phi_y(s)$ (Lines 4–5). If the travelling cost of solution s' is a new local optimum, then the local optimum s^* is updated and y is re-initialised (Lines 6–7). Otherwise, y is incremented and the corresponding neighbourhood is evaluated (Lines 8–9). The algorithm terminates if no improving solution can be obtained when exploring the last neighbourhood y_{max} (Line 3).

Algorithm 4: VND algorithm

Input: s
//Set of neighbourhood moves, $y = 1, 2, \ldots, y_{max}$//
$s^* \leftarrow \emptyset$
$s^* \leftarrow s$
for $y \leftarrow 1$ to y_{max} **do**
  $\Phi_y(s) \leftarrow$ Build Allowed Set(s, y)//Neighbourhood evaluation
  $s \leftarrow \arg\min_{s' \in \Phi_y(s)}\{f(s')\}$
  if $f(s^*) > f(s)$ **then**
    $y \leftarrow 1$, $s^* \leftarrow s$
  else
    $y \leftarrow y+1$
  end if
end for
Output: s^*

From the implementation viewpoint, neighbourhood change and selection is applied to the following order: 2-Opt, Swap(1,1) and Shift(1,0). It is worth highlighting that in order to reduce computational effort of the neighbourhood evaluation, only the necessary move combinations are evaluated. Specifically, since VND is mostly applied after HTS has performed a chosen move to a solution, this means that at most three vehicles will have been altered. Thus, it is necessary to evaluate the move combinations that involve the modified routes only.

5. Computational results

For the evaluation of the proposed metaheuristic a number of computational experiments were carried out utilising existing and newly generated benchmark datasets. More specifically, to evaluate the robustness of the developed solution approach, the existing ConVRP benchmark instances were used. Moreover, to further test the performance of the HTS algorithm, 11 new small-scale instances for the HConVRP were generated and solved to optimality using the proposed mathematical model. Lastly, new medium and large-scale HConVRP instances for algorithmic component testing and managerial insight analysis were created.

The proposed HTS framework incorporates four input parameters, i.e. ω_t (tabu list size), I_{TS} (HTS algorithm termination condition) and β and γ (HTS neighbourhood oscillation parameters). A rich scientific literature is devoted to TS algorithm (Gendreau and Potvin, 2019; Vela et al., 2020; Mathlouthi et al., 2021); thus standard settings in the VRP literature are used for the HTS with $\omega_t = 50$ and $I_{TS} = 30\,000$, providing a good balance between effectiveness and computational time consumption. Moreover, β and γ are set to 50 and 75, respectively, as discussed in Section 4.4. As far as the multi-start mechanism parameter Θ is concerned, the formula $\frac{n}{10}$ is utilised, where n is the number of customers of the instance (Stavropoulou et al., 2019).

All experimental results reported in the following sections consider the aforementioned fixed parameter settings over five runs (apart from Section 5.2 in which 10 runs were conducted for fairness in comparison). The average and best results of these runs are reported, denoted as TT_{avg} and TT_{min} respectively, along with the average computational times, denoted as CT. The proposed metaheuristic was coded in Java and all computational experiments were performed on a 3.30 GHz Intel Core i5-4590 PC over a single thread.

5.1. Benchmark datasets

The existing ConVRP benchmark dataset (Dataset A) consists of 12 medium and large-scale problem instances, containing 50 to 199 customers, divided into two groups. The first group contains seven problem instances, considering vehicle capacity constraints only. The second group includes both vehicle capacity and route duration constraints. A 5-day planning horizon with an unlimited fleet size is assumed. It should be highlighted that the objective function values refer to the sum of travelling distance and customer service times, i.e. the total en route time. To follow the proposed algorithm's rationale, one vehicle type was assumed, i.e. $H = 1$, with $F_1 = 0$, $V_1 = 1$ and $B_1 = 1$. The available fleet size h_1 was set equal to the fleet size reported in Tarantilis et al. (2012). To be consistent with the literature practice and to ensure a fair comparison with the state-of-the-art, the maximum arrival time difference limit L was bounded based on the results of Groër et al. (2009).

To generate the HConVRP benchmark instances, the existing ConVRP instances were adapted. The small-scale ConVRP instances contain up to 12 customers. A 3-day planning horizon is assumed and vehicle capacity and route duration are constrained. The number of available vehicles is unlimited, therefore, a fleet size and composition was defined experimentally. Additionally, in some instances, a few frequent customers became non-frequent, as the initial number of non-frequent customers was too small. Lastly, the small-scale instances set was enriched by generating larger instances including 18 customers, adopting the same rules (Dataset B).

Following the same process, the medium and large-scale HConVRP benchmark instances were created (Dataset C). The fleet size and composition was determined experimentally and the number of non-frequent customers were altered accordingly. Thus, the medium and large-scale HConVRP benchmark instances are grouped into three different sets according to the percentage of non-frequent customers they include. The first set contains the smallest percentage of non-frequent customers (15%)(set 1), in the second set 25% of customers are non-frequent (set 2) whilst the third set contains the largest percentage of non-frequent customers (50%)(set 3). As far as the fleet composition is concerned, the same rationale as in Duhamel et al. (2011) regarding the correlation between capacity and fixed and variable costs was followed. In this case, the vehicle capacities are uncorrelated to vehicle costs. For example, the larger fixed costs to purchase a hybrid vehicle are offset by smaller operating costs.

Lastly, to create the customer–vehicle incompatible HConVRP benchmark instances (Dataset D), the aforementioned medium and large-scale HConVRP benchmark instances were modified. In particular, a scenario where 25% of the customers require to be serviced by vehicles of a certain type was adopted. The remaining customers do not have any requirements, hence they can be visited by any vehicle type. In the adopted scenario, 50% of the customers imposing vehicle type compatibility constraints require to be visited by smaller vehicles (i.e. the vehicle type with the smallest capacity), while the remaining 50% require to be visited by larger vehicles (i.e. the vehicle type with the largest capacity). Customers that need to be serviced by smaller vehicles could be customers located in urban areas with narrow streets and limited parking space, whilst customers that need to be serviced by larger vehicles could be commercial customers demanding large delivery quantities located in suburban areas.

5.2. Results for ConVRP benchmark instances (Dataset A)

This section presents the results obtained from the proposed algorithm on the Dataset A instances of Groër et al. (2009) and compares them with the state-of-the-art. Table 1 shows the corresponding results and computational times (in seconds). As there is a number of metaheuristics that have been used to tackle the ConVRP instances and due to lack of space, HTS algorithm is compared with the two best-performing approaches, i.e. the LNS-SP metaheuristic of Goeke et al. (2019) and the VNS metaheuristic of Xu and Cai (2018). For each algorithm and each instance, the best and the average result obtained in 10 runs for the LNS-SP and the proposed HTS are reported, denoted as TT_{min} and TT_{avg} respectively. It is noteworthy that as far as

Table 1
Comparative analysis on ConVRP instances.

#	BKS	LNS-SP					VNS			HTS				
		TT_{min}	TT_{avg}	CT[a]	Δz_a	Δz_b	TT_{avg}	CT[b]	Δz_b	TT_{min}	TT_{avg}	CT[c]	Δz_a	Δz_b
P1	2121.84	2124.21	2128.46	40.50	0.11	0.31	2125.29	8.02	0.16	2135.70	2135.70	23.81	0.65	0.65
P2	3481.72	3494.77	3508.93	86.90	0.37	0.78	3529.01	11.23	1.36	**3481.72**	3489.48	83.49	0.00	0.22
P3	3278.36	**3278.36**	3296.87	195.30	0.00	0.56	3299.85	26.76	0.66	3308.50	3332.76	124.16	0.92	1.66
P4	4355.47	4410.68	4464.36	369.70	1.27	2.50	4433.74	42.51	1.80	4406.30	4434.55	175.19	1.17	1.82
P5	5480.00	5485.78	5581.53	477.60	0.11	1.85	5570.81	72.71	1.66	5529.40	5562.36	310.25	0.90	1.50
P6	4051.48	**4051.48**	**4051.48**	31.70	0.00	0.00	**4051.48**	8.12	0.00	4051.65	4051.65	16.36	0.00	0.00
P7	6645.05	**6645.05**	**6645.05**	73.60	0.00	0.00	6667.12	14.22	0.33	6652.06	6660.50	57.35	0.11	0.23
P8	7094.05	**7094.05**	7119.16	145.50	0.00	0.35	7132.48	28.25	0.54	7105.90	7116.65	102.85	0.17	0.32
P9	10 318.99	10 329.99	10 371.52	367.60	0.11	0.51	10 389.90	53.40	0.69	10 341.26	10 356.49	189.85	0.22	0.36
P10	12 839.78	**12 839.78**	12 968.06	467.00	0.00	1.00	12 927.80	85.68	0.69	12 862.32	12 886.59	231.68	0.18	0.36
P11	4447.45	**4447.45**	4493.58	227.10	0.00	1.04	4454.47	57.83	0.16	4458.48	4470.35	108.03	0.25	0.51
P12	3416.08	**3416.08**	3427.97	125.90	0.00	0.35	3489.07	16.41	2.14	3421.89	3431.60	95.69	0.17	0.45
Average	5627.52	5634.81	5671.41	217.37	0.16	0.77	5672.59	35.43	0.85	5646.27	5660.72	126.56	0.39	0.68

[a]AMD FX-6300 3.50 GHz.
[b]Intel Core i5-6500 3.40 GHz.
[c]Intel Core i5-4590 3.30 GHz.

the VNS approach is concerned, the authors report only the average results obtained in 10 runs. Therefore, the results presented in this table are denoted as TT_{avg}. The best-known solution in the literature is denoted as BKS, the percentage gap of the best solution found by each algorithm to BKS is denoted as Δz_a, the percentage gap of the reported average solution value to the BKS is denoted as Δz_b, and the average computational time in seconds is denoted as CT. The average results over all instances are given in the last row of the table. The best-known solutions are indicated in bold.

As illustrated in Table 1, the proposed HTS algorithm performed well, compared to the state-of-the-art, and found one best-known solution. As far as the computational time is concerned, the HTS framework required a competitive computational time, proving its efficiency. According to Cordeau et al. (2002), VRP heuristics should possess or be compared using four criteria: accuracy, speed, simplicity and flexibility. Accuracy refers to the gap between the obtained heuristic solution and optimal value (if existing) or best-known solutions, simplicity refers to the ease of implementation related to a low number of parameters and flexibility is related to the ability of an algorithm to accommodate various side constraints. As far as accuracy and speed are concerned, the proposed metaheuristic provides a very good combination of solution quality and speed. In terms of simplicity, HTS is a simple and easy to implement framework as it incorporates only four user-defined parameters and many deterministic components with randomness deriving only from the solution construction heuristic, while both LNS-SP and VNS utilise a number of parameters and several algorithmic components that introduce randomness. Last, HTS is a flexible metaheuristic, accommodating various side constraints, as shown in the computational experiments presented below. All the above indicate that the HTS is a competitive solution method for the ConVRP.

5.3. Results for HConVRP benchmark instances

In this section all the computational results obtained on the new small, medium and large-scale benchmark instances for the HConVRP are discussed. In particular, Dataset B instances are solved to optimality using a commercial solver and the proposed algorithm's results are compared to the optimal solutions. Moreover, Dataset C instances are used to test the efficiency of the developed algorithmic components as well as to examine the trade-off between the total transportation cost and the service consistency constraints and customer–vehicle compatibility constraints. In the latter experiments, apart from the total transportation cost, the focus is on the provided service level, i.e. the maximum arrival time difference (L_{max}), as low values of this metric result in "more consistent" daily schedules.

5.3.1. Small-scale instances (Dataset B)

To further test the performance of the proposed metaheuristic, the generated Dataset B instances were solved to optimality, using a commercial solver (ILOG CPLEX 12.7). Table 2 shows the optimal solutions and the results obtained by the HTS framework. The optimal objective function value (TT), the root node, the computational time in seconds (CT) and the number of nodes examined during the CPLEX execution are reported for each optimal solution. The best and average obtained objective function value over five runs, denoted TT_{min} and TT_{avg}, respectively, are reported for the proposed metaheuristic. The percentage gap of the best and average objective function values obtained by HTS compared to the optimal values, i.e. Δz_a and Δz_b respectively, are calculated. In all cases HTS took less than a second to find the presented solutions, thus the corresponding computational times are not reported.

As shown in Table 2, the proposed HTS framework managed to acquire high quality solutions. Specifically, it obtained six optimal solutions, while its average deviation from the optimal objective function values was 0.26% (worst case performance is 1.51%). As far as its average performance is concerned, the proposed solution method obtained an average deviation from the optimal values of 0.82%, with a worst case performance of 2.16%.

Apart from solving the HConVRP version, different scenarios with different combinations of constraints were executed to further evaluate the effectiveness of the proposed metaheuristic, as well as examining the trade-off of service consistency constraints (discussed in Section 5.4). In particular, three scenarios were adopted. The first scenario, shown in Table 3, imposes only driver consistency constraints, whereas the second scenario, presented in Table 4, assumes only arrival time consistency constraints. In the third scenario, reported in Table 5, all service consistency constraints are inactive; in this case the HConVRP is reduced to a multi-period HVRP. The aforementioned tables use the same notation discussed in Table 2.

As far as the first scenario is concerned, constraints (12) of the mathematical model, i.e. the constraints dictating arrival time consistency, became inactive. Table 3 shows that HTS found the optimal solution in 9 out of 11 cases, with an average deviation from the optimal objective function values of 0.25% (worst case performance is 2.79%). In addition, concerning the proposed metaheuristic's average performance, it obtained five optimal solutions and an average deviation from the optimal values of 0.53%, with a worst case performance of 4.4%.

In the second scenario, constraints (5) of the mathematical model, i.e. the driver consistency constraints, were inactive. As shown in Table 4, the proposed HTS framework managed to acquire three optimal solutions, while its average deviation from the optimal objective function values was 0.36% (worst case performance is 1.1%). As far

Table 2
Comparative analysis on HConVRP small-scale instances.

#	CPLEX				HTS			
	TT	Root node	CT	# Nodes	TT_{min}	TT_{avg}	Δz_a	Δz_b
P1	1162.16	1041.77	15.00	7392	1162.19	1170.94	0.00	0.76
P2	1115.06	1004.32	183.47	97 921	1115.06	1116.51	0.00	0.13
P3	1028.34	912.96	97.47	54 181	1028.34	1029.65	0.00	0.13
P4	1182.40	1097.07	37.19	11 744	1182.40	1182.40	0.00	0.00
P5	1113.27	1002.85	277.72	108 571	1113.27	1118.95	0.00	0.51
P6	1258.32	1039.43	771.17	227 460	1258.65	1267.79	0.03	0.75
P7	1076.83	963.91	219.49	111 977	1076.83	1080.92	0.00	0.38
P8	1101.50	1036.44	60.38	24 232	1101.50	1120.82	0.00	1.75
P9	1240.51	1110.10	114.59	57 719	1246.30	1252.46	0.47	0.96
P10	1169.56	1020.06	173.06	59 797	1187.24	1194.85	1.51	2.16
P11	1183.45	1087.50	284 770.64	64 930 379	1193.62	1200.53	0.86	1.44
Average	1148.31	1028.76	26 065.47	5 971 943	1151.40	1157.80	0.26	0.82

Table 3
Comparative analysis on HConVRP small-scale instances (only driver consistency).

#	CPLEX				HTS			
	TT	Root node	CT	# Nodes	TT_{min}	TT_{avg}	Δz_a	Δz_b
P1	1162.16	1041.77	31.42	23 137	1162.19	1162.19	0.00	0.00
P2	1114.83	1004.32	239.91	275 888	1114.83	1116.69	0.00	0.17
P3	1023.26	912.96	32.08	23 494	1051.85	1068.72	2.79	4.40
P4	1182.40	1097.06	112.59	61 379	1182.40	1182.40	0.00	0.00
P5	1108.61	1002.85	195.44	102 297	1108.61	1118.58	0.00	0.90
P6	1257.97	1039.43	1386.69	940 462	1257.97	1258.05	0.00	0.01
P7	1076.79	963.91	313.26	217 746	1076.79	1076.79	0.00	0.00
P8	1100.45	1036.44	73.36	52 129	1100.45	1100.45	0.00	0.00
P9	1236.86	1110.10	1768.94	831 942	1236.86	1240.39	0.00	0.29
P10	1151.82	1020.06	141.44	112 973	1151.82	1151.82	0.00	0.00
P11	1181.94	1087.50	561 827.75	181 146 805	1181.94	1181.94	0.00	0.00
Average	1145.19	1028.76	51 465.72	16 708 022.91	1147.79	1150.73	0.25	0.53

Table 4
Comparative analysis on HConVRP small-scale instances(only arrival consistency).

#	CPLEX				HTS			
	TT	Root node	CT	# Nodes	TT_{min}	TT_{avg}	Δz_a	Δz_b
P1	1133.09	1033.25	24.08	15 075	1137.28	1140.13	0.37	0.62
P2	1091.88	1004.32	239.91	275 888	1096.95	1106.69	0.46	1.36
P3	986.31	902.98	55.30	34 165	991.48	992.43	0.52	0.62
P4	1164.28	1090.42	1557.51	676 967	1164.28	1172.72	0.00	0.73
P5	1104.53	1001.17	7872.77	3 126 817	1110.45	1111.72	0.54	0.65
P6	1238.65	1033.12	11 496.25	4 142 312	1252.24	1253.99	1.10	1.24
P7	1050.86	954.55	350.51	173 024	1050.86	1052.26	0.00	0.13
P8	1094.89	1025.54	1312.97	619 065	1094.89	1094.89	0.00	0.00
P9	1219.30	1104.06	5431.05	2 948 275	1226.08	1231.42	0.56	0.99
P10	1158.09	1010.91	8439.44	4 747 064	1158.73	1160.68	0.06	0.22
P11	1228.74*	1068.72	394 482.91	73 767 236	1192.23	1204.27	(−2.97)	(−1.99)
Average	1133.69	1020.82	39 205.70	8 229 626.18	1134.13	1138.29	0.36	0.66

as its average performance is concerned, the proposed solution method obtained an average deviation from the optimal values of 0.66%, with a worst case performance of 1.36%. It should be noted that, in this scenario, CPLEX was unable to find the optimal solution for P11. Thus, the objective function value reported for P11 (highlighted with an asterisk) is the MIP solution obtained by CPLEX, with an optimality gap of 8.28%. For this reason, the objective function value of the corresponding HTS solution is lower than the one obtained by CPLEX. It is worth mentioning that Δz_a and Δz_b for P11 are not included in the average deviation calculation presented in the last row of Table 4.

In the last scenario, constraints (5) and (12) of the mathematical model, i.e. both the driver and arrival time consistency constraints, became inactive. Table 5 shows that HTS found six optimal solutions, with an average deviation from the optimal objective function values of 0.22% (worst case performance is 0.87%). In addition, concerning the proposed metaheuristic's average performance, it obtained five optimal solutions and an average deviation from the optimal values of 0.47%, with a worst case performance of 2.2%.

All the aforementioned results demonstrate the effectiveness of the proposed HTS framework and its flexibility to adapt to different combinations of service consistency constraints.

5.3.2. Medium and large-scale instances (Dataset C)

As shown above, the computational time for solving instances of realistic size to optimality is excessive. For this reason, larger instances were solved using the HTS metaheuristic framework. A number of computational experiments were carried out on these instances to test the performance of the proposed algorithmic mechanisms.

Initially, the contribution of the hierarchical scheme is examined in terms of the effectiveness (obtained solution quality), service consistency and efficiency (required computational effort). More specifically, various computational experiments were conducted without invoking the VND algorithm to perform the lower level search, aiming to optimise the routing of the customers (denoted as HTS-noVND). Additionally, the proposed deterministic oscillation of neighbourhoods of the HTS algorithm is compared to the "common" random oscillation of neighbourhood structures adopted in most TS implementations

Table 5
Comparative analysis on multi-period HVRP small-scale instances.

#	CPLEX				HTS			
	TT	Root node	CT	# Nodes	TT_{min}	TT_{avg}	Δz_a	Δz_b
P1	1132.60	1073.73	1.55	3368	1132.60	1132.60	0.00	0.00
P2	1089.39	1003.82	795.84	740 291	1094.97	1094.97	0.51	0.51
P3	980.72	921.45	7.50	11 534	980.72	1002.28	0.00	2.20
P4	1159.82	1098.49	5.20	6597	1159.82	1159.82	0.00	0.00
P5	1064.67	1017.88	9.47	14 609	1073.04	1076.14	0.79	1.08
P6	1236.00	1033.12	22 892.20	24 859 699	1238.24	1240.76	0.18	0.39
P7	1050.86	972.91	11.94	14 156	1050.86	1050.86	0.00	0.00
P8	1087.01	1034.54	10.66	12 201	1087.01	1087.01	0.00	0.00
P9	1214.08	1104.06	42.03	50 261	1214.08	1214.08	0.00	0.00
P10	1141.54	1030.06	29.69	37 555	1151.46	1151.46	0.87	0.87
P11	1168.47	1068.72	459 788.73	159 011 177	1168.82	1169.62	0.03	0.10
Average	1120.47	1032.62	43 963.16	16 796 495.27	1122.87	1125.42	0.22	0.47

(denoted as TS). It is worth highlighting that both HTS-noVND and TS use the same fixed parameter settings with HTS to ensure fairness in their comparison. Moreover, it is noteworthy that in all the experiments presented in this section, L is not bounded. The corresponding computational results are summarised in Table 6. For each algorithm and each instance, the average total transportation cost (TT_{avg}), the average maximum arrival time difference (L_{max}) and the average computational time in seconds (CT) over five runs are reported. Finally, the %Gap between HTS framework and HTS-noVND version, denoted as $\%Gap_1$, along with the %Gap between HTS algorithm and TS algorithm, denoted as $\%Gap_2$, were calculated.

As far as the contribution of the hierarchical search framework is concerned, in all cases the proposed HTS algorithm proved superior compared to the corresponding single-level HTS-noVND scheme (without the VND component). In particular, the HTS metaheuristic outperformed the HTS-noVND algorithm in terms of the obtained solutions' objective function value, with an average deviation of 12.26%, achieving improvements of up to 38.57%. The highest average deviation was observed in the problem set with the highest percentage of non-frequent customers (set 3). This demonstrates that the routing counterpart of the HConVRP becomes more crucial as the number of non-frequent customers increases. Thus, a strategy of effectively routing the customers within the daily schedules can lead to high quality solutions. As shown in Table 6, both algorithms obtained similar service consistency, i.e. in terms of the average L_{max}, and required comparable computational times, with the HTS being faster than the HTS-noVND. All the aforementioned demonstrate that the contribution of the proposed hierarchical search framework is major, improving significantly the overall performance and expediting the overall search process in terms of computational times.

Another interesting point of investigation is the oscillation of neighbourhood structures within the HTS component implementation. In the proposed HTS algorithm, the neighbourhood oscillation is deterministic, while TS algorithm implementations commonly adopt a random oscillation of neighbourhood structures. For this reason, several computational experiments were performed to evaluate both implementations. All in all, comparable results were obtained by the HTS and TS frameworks, indicating the effectiveness of the proposed HTS metaheuristic. Specifically, HTS obtained slightly better solutions in terms of the average objective function value than the TS algorithm, with improvements of up to 6.84%. Furthermore, HTS was much faster than the TS algorithm, requiring 10 times less computational time to acquire its solutions. The above indicate that even though, in the case of TS, more computational time was devoted in terms of exploring more complex neighbourhoods, the acquired results did not reflect this. These findings are in line with the literature. Penna et al. (2019) developed a hybrid heuristic to solve a broad class of HVRPs and reported that optimising systematically the vehicle assignment decisions, in combination with changes of sequences, did not yield large solution improvements, as one might have expected. Thus, a more intensive search on classical neighbourhoods seems to be more efficient than an extensive search for alternative fleet assignment in the HVRP context. Finally, the HTS solutions were "more consistent" than the TS ones by 80.22%. This is due to the fact that if the routing and the vehicle assignment of a number of frequent customers is constantly altered, without restraining L, then the daily schedules become "less consistent" in terms of the average obtained L_{max}. Overall, it can be concluded that the hierarchical design of the proposed framework, incorporating a number of deterministic components, guides and accelerates the search process towards high quality solutions.

5.4. Cost of service consistency

In this section, the price of service consistency is discussed based on computational results using the generated Dataset B and Dataset C instances. More specifically, the optimal total transportation costs for the Dataset B instances following different scenarios with different combinations of constraints are presented in Section 5.4.1 to examine the cost of service consistency.

Moreover, the trade-off of service consistency is further evaluated utilising the Dataset C instances. In particular, the effect of decreasing the maximum arrival time difference, i.e. providing more consistent customer service, on the total transportation cost, the fixed cost and the variable cost is examined. Thus, various computational experiments were carried out limiting the value of L. To determine reasonable values for L, HTS was run on each instance without constraining L. The vector of L_{max} found during the aforementioned runs is indicated as L_1. The vectors corresponding to the reduced L values are indicated as $L_{0.8}$, $L_{0.6}$, $L_{0.4}$ and $L_{0.2}$ and are computed by multiplying L_1 by 0.8, 0.6, 0.4 and 0.2, respectively. It should be highlighted that in this paper no idling is allowed either at the depot or at any customer's location. Thus, tightening the arrival time difference cannot be overcome by shifting the vehicles' departure times or by increasing the number of vehicles, since the fleet size is fixed, as in Kovacs et al. (2014b), resulting in a cost increase, presented in Section 5.4.2.

5.4.1. Small-scale instances

To initially evaluate the trade-off of service consistency, different scenarios with different combinations of constraints were solved to optimality. Fig. 4 shows the obtained computational results. As expected, imposing consistency constraints results in an increase of the total transportation cost. In particular, when applying arrival time consistency constraints there is a 0.76% increase in the total transportation cost compared to the cost of multi-period HVRP, whereas when imposing driver consistency constraints the increase in the total transportation cost is 2.32%. Finally, when combining both arrival time and driver consistency constraints, i.e. in the HConVRP version, the cost increase rises to 2.61%. It is noteworthy that in the scenario that only driver consistency constraints are active, a minimal difference from the HConVRP solution cost is observed. This is due to the fact that driver

Table 6
Results on HConVRP medium and large-scale instances.

#	HTS			HTS-noVND			TS			$\%Gap_1$	$\%Gap_2$
	TT_{avg}	L_{max}	CT	TT_{avg}	L_{max}	CT	TT_{avg}	L_{max}	CT		
P1	11 417.35	134.99	46.20	11 841.84	134.70	32.46	11 518.38	87.02	687.39	−3.72	−0.88
P2	11 338.74	168.22	115.68	12 659.15	160.85	151.33	11 366.00	213.98	358.83	−11.65	−0.24
P3	10 722.74	129.80	133.39	11 358.74	126.48	231.95	10 503.93	310.47	3492.12	−5.93	2.04
P4	15 840.05	136.74	210.30	16 646.72	135.70	248.40	15 473.77	157.58	1240.39	−5.09	2.31
P5	19 867.96	98.98	504.57	20 891.16	119.77	373.76	19 859.19	267.23	7162.74	−5.15	0.04
P6	12 096.82	245.77	63.94	12 675.80	225.74	16.51	12 238.79	140.82	514.74	−4.79	−1.17
P7	12 735.75	260.07	118.17	13 067.09	218.53	99.78	12 016.62	474.49	995.72	−2.60	5.65
P8	10 712.07	265.69	151.68	11 356.33	249.31	441.50	11 001.62	250.13	8164.84	−6.01	−2.70
P9	14 665.99	278.57	649.68	15 749.04	245.55	1201.65	15 005.47	280.10	4714.78	−7.38	−2.31
P10	19 563.84	277.40	1345.24	20 544.80	236.68	813.55	19 414.82	353.34	5926.73	−5.01	0.76
P11	13 787.72	205.18	434.77	15 081.97	212.93	602.94	13 929.81	221.45	6337.08	−9.39	−1.03
P12	13 462.23	128.12	207.46	14 430.29	135.34	282.42	13 518.29	238.18	1159.56	−7.19	−0.42
Average(set 1)	13 850.94	194.13	331.76	14 691.91	183.47	374.69	13 820.56	249.57	3396.24	−6.16	0.17
P13	9628.27	123.42	16.06	10 219.27	145.50	17.54	9647.58	226.17	301.60	−6.14	−0.20
P14	10 995.86	169.39	101.40	11 813.62	152.38	47.91	10 885.61	232.98	874.18	−7.44	1.00
P15	9873.55	141.49	61.12	10 403.28	145.48	127.75	9901.57	273.04	2002.68	−5.37	−0.28
P16	14 518.13	136.22	409.48	15 571.73	158.29	563.56	14 612.74	252.08	6591.88	−7.26	−0.65
P17	17 889.12	124.61	1052.83	19 286.04	150.29	1184.74	17 846.94	311.71	12 552.35	−7.81	0.24
P18	10 035.62	224.21	34.81	11 041.55	206.57	17.28	10 076.87	251.23	289.75	−10.02	−0.41
P19	11 078.79	233.46	129.69	13 006.85	221.98	58.19	11 025.67	240.49	473.95	−17.40	0.48
P20	10 572.45	264.94	225.61	11 678.05	230.80	304.22	10 408.36	308.65	1006.40	−10.46	1.55
P21	14 204.24	239.30	627.21	15 382.26	213.85	1060.88	14 155.24	319.89	9507.97	−8.29	0.34
P22	18 063.81	251.77	1549.95	20 630.05	241.91	1937.02	18 217.70	362.56	12 260.48	−14.21	−0.85
P23	11 625.48	234.99	400.90	13 665.89	195.37	310.08	11 290.49	1064.35	2370.56	−17.55	2.88
P24	12 973.11	153.30	144.54	14 034.27	126.07	84.94	12 841.18	302.95	940.77	−8.18	1.02
Average(set 2)	12 621.54	191.43	396.13	13 894.41	182.38	476.18	12 575.83	345.51	4097.71	−10.01	0.43
P25	9094.71	127.24	21.87	10 178.14	171.92	3.64	9094.80	564.11	178.85	−11.91	0.00
P26	9810.23	184.60	77.85	11 236.92	174.98	11.14	9897.76	358.03	637.24	−14.54	−0.89
P27	7933.24	145.67	85.09	9311.37	175.16	31.53	8223.13	347.91	532.46	−17.37	−3.65
P28	11 794.33	149.28	315.39	13 473.98	197.16	85.65	12 031.97	408.79	2778.97	−14.24	−2.01
P29	14 361.33	120.59	998.28	16 654.13	159.51	527.06	14 446.67	293.70	5670.96	−15.97	−0.59
P30	8086.79	180.49	45.78	11 206.00	174.40	7.85	8639.76	329.63	45.37	−38.57	−6.84
P31	9324.83	207.73	54.04	11 931.87	183.82	28.42	9873.35	325.20	58.74	−27.96	−5.88
P32	8546.57	251.94	72.24	10 078.29	272.66	68.25	9084.90	504.04	1094.39	−17.92	−6.30
P33	11 660.87	231.94	290.17	14 758.12	199.39	395.07	11 735.59	356.77	1326.92	−26.56	−0.64
P34	14 174.31	246.12	467.61	17 178.95	217.76	344.13	14 131.08	737.71	5393.03	−21.20	0.30
P35	10 374.90	232.63	274.63	12 601.46	226.26	84.42	10 541.72	607.17	701.53	−21.46	−1.61
P36	10 684.56	159.60	65.82	12 787.91	187.17	26.21	10 628.45	397.51	334.68	−19.69	0.53
Average(set 3)	10 487.22	186.49	230.73	12 616.43	195.02	134.45	10 694.10	435.88	1562.76	−20.62	−2.30
Average	12 319.90	190.68	319.54	13 734.25	186.95	328.44	12 363.50	343.65	3018.91	−12.26	−0.57

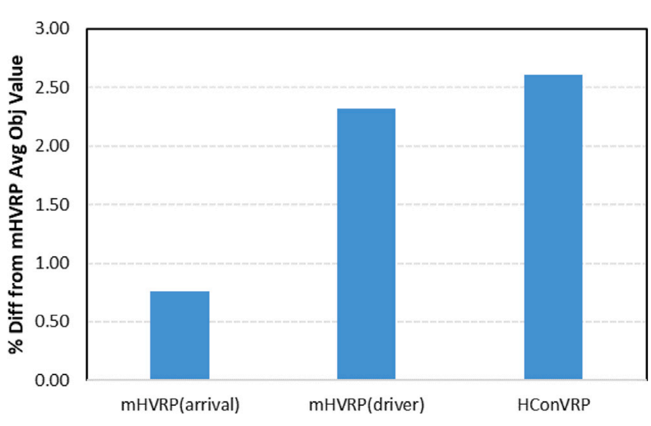

Fig. 4. Cost of service consistency - small-scale instances.

consistency leads to solutions following the template route rationale. Therefore, the solutions obtained by imposing only driver consistency constraints are quite similar to the ones obtained when both driver and arrival time consistency constraints are followed.

5.4.2. Cost of arrival time consistency

After analysing the effect of the different consistency constraints to the total transportation cost using the Dataset B instances, a number of computational experiments were conducted utilising the Dataset C instances. Figs. 5–7 present the obtained computational results.

Fig. 5 summarises the effect of decreasing L on the average objective function value, i.e. the total transportation cost. As expected, restricting the value of L leads to increased transportation costs. As depicted in Fig. 5, decreasing L by 60% results in an average transportation cost increase of 1.27%, whereas further decreasing L by 20% leads to a further cost increase of 4.04%. This means that improving service consistency by 60% costs 1.27% in the total transportation costs, while improving service consistency by 80% costs 5.31% in the total transportation costs.

Fig. 6 illustrates the effect of offering better service consistency on the average fixed costs. It is clear that constraining the value of L leads to increased fixed costs. In particular, Fig. 6 indicates that decreasing L by 60% causes an average increase of 0.42% in the fixed costs, while decreasing L by 80% causes an average increase of 1.22%. This means that improving the offered service consistency by 60% is followed by an increase of 0.42% in fixed costs, whereas improving service consistency by 80% costs 1.22% in fixed costs. It should be highlighted that the results concerning $L_{0.6}$ and $L_{0.4}$ demonstrate a smaller increase in the fixed costs compared to the $L_{0.8}$ results. This is due to the fact that by further constraining the arrival time, more customers have been

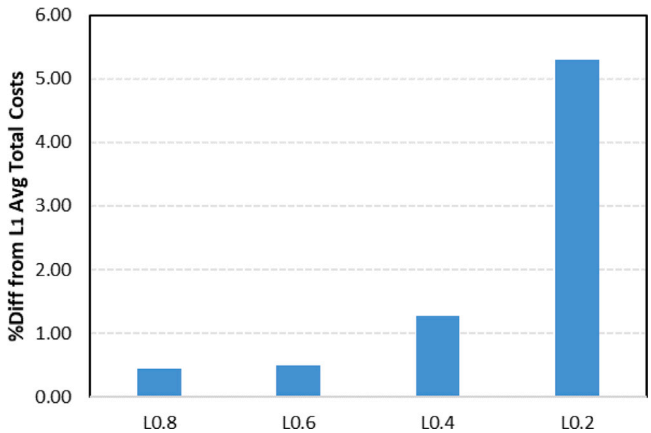

Fig. 5. Effect of L value on the total transportation cost.

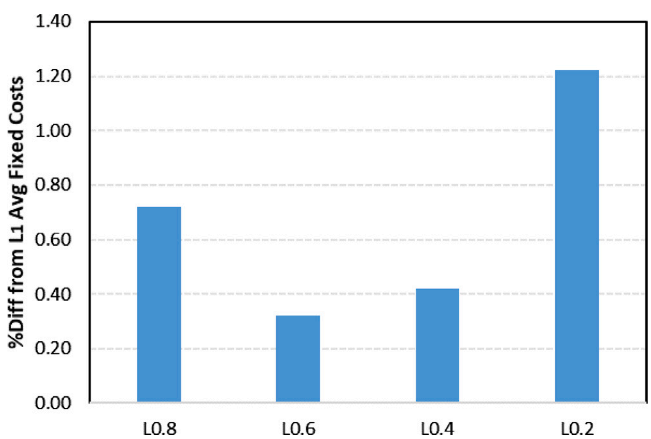

Fig. 6. Effect of L value on the fixed cost.

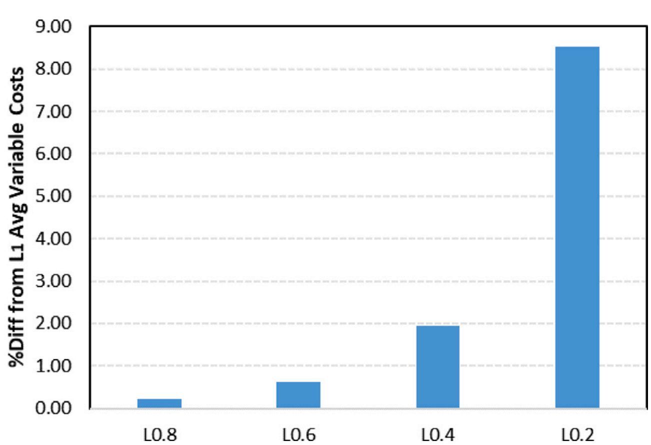

Fig. 7. Effect of L value on the variable cost.

grouped and assigned to specific vehicles, resulting in the utilisation of fewer vehicles on certain days of the planning horizon. For this reason, the increase of the fixed costs was smaller in these cases.

Fig. 7 depicts the impact of improving service consistency on the average variable costs. Following the same trend as the total transportation costs, limiting L by 60% results in an average increase of 1.95% in the variable costs, whilst limiting L by 80% results in an average increase of 8.53%. In other words, an improvement on service consistency of 60% costs 1.95% in variable costs, whereas improving service consistency by 80% costs 8.53% in variable costs.

Table 7
Cost of customer–vehicle incompatibility.

% Diff from dataset C instances				
	TT	VC	FC	L_{max}
Set 1	5.63	10.39	−0.33	−4.72
Set 2	7.87	12.79	1.84	−2.12
Set 3	3.74	4.36	2.89	−1.72
Overall	5.85	9.42	1.31	−2.88

These findings agree with the literature. According to Emadikhiav et al. (2020), more consistent schedules can be obtained with a modest increase in transportation costs (increase from 1% up to 2.2%). Stavropoulou et al. (2019) study the ConVRP with profits and report that improving arrival time consistency by 60% results in an average increase of 6.18% in the travelling costs. Subramanyam and Gounaris (2016) demonstrate that in order to provide consistent service in the ConTSP context, cost is increased by 1.31% on average. Additionally, while examining the multi-objective version of the ConVRP, Kovacs et al. (2015b) showed that improving arrival time consistency by 70% leads to an average increase in the travelling distance of 2.43%.

Overall, the findings presented above indicate that decision-makers should carefully consider the service consistency level that will be offered to customers as this can have an impact on overall transportation costs, imposing a negative effect on profitability.

5.5. Cost of customer–vehicle incompatibility

In this section, the cost of customer–vehicle incompatibility constraints is examined. In particular, the effect of imposing customer–vehicle incompatibility restrictions on the total transportation cost, the variable cost, the fixed cost and L_{max} is discussed. Therefore, several computational experiments were performed, utilising the generated Dataset D instances. Since these instances reflect the scenario that certain customers require to be serviced only by specific vehicle types, this resulted in cases where the constructive heuristic was unable to find an initial solution. For this reason, the constructive heuristic was modified to prioritise the routing of the customers with compatibility constraints. However, for some instances this was still insufficient and the fleet size had to be increased. It should be highlighted that both driver and arrival time consistency constraints were active for these experiments. The obtained computational results are summarised in Table 7. Specifically, Table 7 presents the % Gap between the solutions obtained by HTS on the Dataset C and Dataset D instances in terms of the average total transportation cost (TT), the average variable cost (VC), the average fixed cost (FC) and the average L_{max}.

As expected, dictating customer–vehicle compatibility restrictions leads to increased transportation costs. As shown in Table 7, the average total transportation cost is increased by 5.85%, the average variable cost by 9.42%, whereas the average fixed cost is increased by 1.31%. However, customer–vehicle restrictions seem to produce "more consistent" solutions, since the obtained average L_{max} is decreased by 2.88%. This is due to the fact that by imposing customer–vehicle compatibility constraints, more customers have been grouped and assigned to specific vehicles, resulting in producing "more consistent" solutions. For the same reason, fewer vehicles were utilised on certain days of the planning horizon, leading to the average fixed cost decreasing by 0.33% in set 1.

6. Conclusions

Customer relationship management is key as it enhances customer satisfaction and retention. Thus, companies focus on providing consistent customer service in an attempt to form long-term relationships with their customers and cultivate brand loyalty.

A new customer-oriented routing problem was introduced, the Consistent Vehicle Routing Problem with heterogeneous fleet (HConVRP), taking into account a fixed fleet with heterogeneous characteristics, along with person and arrival time consistency. A mathematical model capturing all the attributes of the problem was developed and utilised to solve small-scale instances to optimality. For addressing larger instances, a hierarchical Tabu Search (HTS) framework was proposed, utilising an upper-level Tabu Search and an underlying Variable Neighbourhood Descent algorithm, exploiting different search landscapes. For the performance evaluation of the proposed framework, existing and newly generated benchmark instances were utilised, proving its flexibility, effectiveness and efficiency. Overall, it was shown that the hierarchical design of the HTS framework, incorporating a number of deterministic components, guides and accelerates the search process towards high quality solutions.

Our computational study indicates that improving the offered arrival time consistency by 80% leads to an increase of 5.31% in the total transportation cost, an increase of 1.22% in fixed costs and an increase of 8.53% in variable costs on average. Furthermore, when customers impose vehicle compatibility constraints, the total transportation cost is increased by 5.85%, the variable cost by 9.42%, whereas the fixed cost is increased by 1.31% on average. All these are valuable managerial insights demonstrating that decision-makers should consider carefully the level of service consistency offered to customers as this may have a negative impact on the costs.

One research direction worth pursuing would be further enriching the problem by incorporating additional realistic attributes, such as dynamic customer requests and stochastic demand or time-dependent travel times.

CRediT authorship contribution statement

F. Stavropoulou: Conceptualisation, Methodology, Software, Formal analysis, Writing - original draft, Writing - review & editing, Visualisation.

Acknowledgements

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

Baldacci, R., Battarra, M., Vigo, D., 2008. Routing a heterogeneous fleet of vehicles. In: Golden, B.L., Raghavan, S., Wasil, E.A. (Eds.), The Vehicle Routing Problem: Latest Advances and New Challenges. Springer, New York, pp. 1–25.

Baldacci, R., Toth, P., Vigo, D., 2010. Exact algorithms for routing problems under vehicle capacity constraints. Ann. Oper. Res. 175 (1), 213–245.

Braekers, K., Kovacs, A.A., 2016. A multi-period dial-a-ride problem with driver consistency. Trans. Res. B 94, 355–377.

Bräysy, O., Dullaert, W., Hasle, G., Mester, D., Gendreau, M., 2008. An effective multirestart deterministic annealing metaheuristic for the fleet size and mix vehicle routing problem with time windows. Transp. Sci. 42 (3), 371–386.

Campelo, P., Neves-Moreira, F., Amorim, P., Almada-Lobo, B., 2019. Consistent vehicle routing problem with service level agreements: a case study in the pharmaceutical distribution sector. European J. Oper. Res. 273 (1), 131–145.

Ceselli, A., Righini, G., Salani, M., 2019. A column generation algorithm for a rich vehicle-routing problem. Transp. Sci. 43 (1), 56–69.

Chao, I.M., Golden, B.L., Wasil, E.A., 1998. A new algorithm for the site-dependent vehicle routing problem. In: Woodruff, D.L. (Ed.), Advances in Computational and Stochastic Optimization, Logic Programming, and Heuristic Search. Springer, Boston, pp. 301–312.

Cordeau, J.-F., Gendreau, M., Laporte, G., Potvin, J.-Y., Semet, F., 2002. A guide to vehicle routing heuristics. J. Oper. Res. Soc. 53, 512–522.

Duhamel, C., Lacomme, P., Prodhon, C., 2011. Efficient frameworks for greedy split and new depth first search split procedures for routing problems. Comput. Oper. Res. 38, 723–739.

Emadikhiav, M., Bergman, D., Day, R., 2020. Consistent routing and scheduling with simultaneous pickups and deliveries. Prod. Oper. Manag. 29 (8), 1937–1955.

Feillet, D., Garaix, T., Lehuédé, F., Péton, O., Quadri, D., 2014. A new consistent vehicle routing problem for the transportation of people with disabilities. Networks 63 (3), 211–224.

Gendreau, M., Potvin, J.-Y., 2019. Tabu search. In: Gendreau, M., Potvin, J.-Y. (Eds.), Handbook of Metaheuristics, third ed. Springer International Publishing, pp. 37–55.

Goeke, D., Roberti, R., Schneider, M., 2019. Exact and heuristic solution of the consistent vehicle-routing problem. Transp. Sci. 53 (4), 1023–1042.

Golden, B.L., Assad, A.A., Levy, L., Gheysens, F., 1984. The fleet size and mix vehicle routing problem. Comput. Oper. Res. 11, 49–66.

Groër, C., Golden, B., Wasil, E., 2009. The consistent vehicle routing problem. Manuf. Serv. Oper. Manag. 11 (4), 630–643.

Hewitt, M., Nowak, M., Gala, L., 2015. Consolidating home meal delivery with limited operational disruption. European J. Oper. Res. 243 (1), 281–291.

Hoff, A., Andersson, H., Christiansen, M., Hasle, G., Løkketangen, A., 2015. Industrial aspects and literature survey: fleet composition and routing. Comput. Oper. Res. 37, 2041–2061.

Ioachim, I., Desrosiers, J., Soumis, F., Bélanger, N., 1999. Fleet assignment and routing with schedule synchronization constraints. European J. Oper. Res. 119 (1), 75–90.

Iori, M., Riera-Ledesma, J., 2015. Exact algorithms for the double vehicle routing problem with multiple stacks. Comput. Oper. Res. 63, 83–101.

Irnich, S., Schneider, M., Vigo, D., 2014. Four variants of the vehicle routing problem. In: Toth, P., Vigo, D. (Eds.), Vehicle Routing: Problems, Methods and Applications. MOS-SIAM Series on Optimization, Philadelphia, pp. 241–271.

Koç, Ç., Bektaş, T., Jabali, O., Laporte, G., 2014. The fleet size and mix pollution-routing problem. Trans. Res. B 70, 239–254.

Koç, Ç., Bektaş, T., Jabali, O., Laporte, G., 2015. A hybrid evolutionary algorithm for heterogeneous fleet vehicle routing problems with time windows. Comput. Oper. Res. 64, 11–27.

Koç, Ç., Bektaş, T., Jabali, O., Laporte, G., 2016. Thirty years of heterogeneous vehicle routing. European J. Oper. Res. 249 (1), 1–21.

Kovacs, A.A., Golden, B.L., Hartl, R.F., Parragh, S.N., 2014a. Vehicle routing problems in which consistency considerations are important: a survey. Networks 64 (3), 192–213.

Kovacs, A.A., Golden, B.L., Hartl, R.F., Parragh, S.N., 2015a. The generalized consistent vehicle routing problem. Transp. Sci. 49 (4), 796–816.

Kovacs, A.A., Parragh, S.N., Hartl, R.F., 2014b. A template-based adaptive large neighborhood search for the consistent vehicle routing problem. Networks 63 (1), 60–81.

Kovacs, A.A., Parragh, S.N., Hartl, R.F., 2015b. The multi-objective generalized consistent vehicle routing problem. European J. Oper. Res. 247 (2), 441–458.

Li, X., Leung, S.C., Tian, P., 2012. A multistart adaptive memory-based tabu search algorithm for the heterogeneous fixed fleet open vehicle routing problem. Expert Syst. Appl. 39 (1), 365–374.

Lian, K., Bennett Milburn, A., Rardin, R.L., 2016. An improved multi-directional local search algorithm for the multi-objective consistent vehicle routing problem. IIE Trans. 48 (10), 975–992.

Liu, F.H., Shen, S.Y., 1999. The fleet size and mix vehicle routing problem with time windows. J. Oper. Res. Soc. 50 (7), 721–732.

Luo, Z., Qin, H., Che, C., Lim, A., 2015. On service consistency in multi-period vehicle routing. European J. Oper. Res. 243 (3), 731–744.

Mancini, S., 2016. A real-life multi depot multi period vehicle routing problem with a heterogeneous fleet: formulation and adaptive large neighborhood search based matheuristic. Trans. Res. C 70, 100–112.

Mancini, S., Gansterer, M., Hartl, R.F., 2021. The collaborative consistent vehicle routing problem with workload balance. European J. Oper. Res. 293 (3), 955–965.

Mathlouthi, I., Gendreau, M., Potvin, J.-Y., 2021. A metaheuristic based on tabu search for solving a technician routing and scheduling problem. Comput. Oper. Res. 125, 105079.

Paraskevopoulos, D.C., Repoussis, P.P., Tarantilis, C.D., Ioannou, G., Prastacos, G.P., 2008. A reactive variable neighborhood tabu search for the heterogeneous fleet vehicle routing problem with time windows. J. Heuristics 14 (5), 425–455.

Penna, P.H.V., Subramanian, A., Ochi, L.S., Vidal, T., Prins, C., 2019. A hybrid heuristic for a broad class of vehicle routing problems with heterogeneous fleet. Ann. Oper. Res. 273 (1–2), 5–74.

Qu, Y., Bard, J.F., 2014. A branch-and-price-and-cut algorithm for heterogeneous pickup and delivery problems with configurable vehicle capacity. Transp. Sci. 49 (2), 254–270.

Rodríguez-Martín, I., Salazar-González, J.-J., Yaman, H., 2019. The periodic vehicle routing problem with driver consistency. European J. Oper. Res. 273 (2), 575–584.

Rusell, D., Rosati, R.J., Rosenfeld, P., Marren, J.M., 2011. Continuity in home health care: is consistency in nursing personnel associated with better patient outcomes? J. Healthc. Qual. 33 (6), 33–39.

Salhi, S., Imran, A., Wassan, N.A., 2014. The multi-depot vehicle routing problem with heterogeneous vehicle fleet: formulation and a variable neighborhood search implementation. Comput. Oper. Res. 52 (Part B), 315–325.

Salhi, S., Wassan, N., Hajarat, M., 2013. The fleet size and mix vehicle routing problem with backhauls: formulation and set partitioning-based heuristics. Trans. Res. E 56, 22–25.

Smilowitz, K., Nowak, M., Jiang, T., 2013. Workforce management in periodic delivery operations. Transp. Sci. 47 (2), 214–230.

Song, Y., Ulmer, M.W., Thomas, B.W., Wallace, S.W., 2020. Building trust in home services - stochastic team-orienteering with consistency constraints. Transp. Sci. 54 (3), 823–838.

Stavropoulou, F., Repoussis, P.P., Tarantilis, C.D., 2019. The vehicle routing problem with profits and consistency constraints. European J. Oper. Res. 274 (1), 340–356.
Subramanyam, A., Gounaris, C.E., 2016. A branch-and-cut framework for the consistent traveling salesman problem. European J. Oper. Res. 248 (2), 384–395.
Subramanyam, A., Gounaris, C.E., 2017. A decomposition algorithm for the consistent traveling salesman problem with vehicle idling. Transp. Sci. 52 (2), 386–401.
Subramanyam, A., Repoussis, P.P., Gounaris, C.E., 2020. Robust optimization of a broad class of heterogeneous vehicle routing problems under demand uncertainty. J. Comput. 32 (3), 661–681.
Sungur, I., Ren, Y., Ordóñez, F., Dessouky, M., Zhong, H., 2010. A model and algorithm for the courier delivery problem with uncertainty. Transp. Sci. 44 (2), 193–205.
Taillard, É.D., 1999. A heuristic column generation method for the heterogeneous fleet vehicle routing problem. RAIRO (Res. Oper.) 33 (1), 1–14.
Tarantilis, C.D., Stavropoulou, F., Repoussis, P.P., 2012. A template-based tabu search algorithm for the consistent vehicle routing problem. Expert Syst. Appl. 39 (4), 4233–4239.
Ulmer, M., Nowak, M., Mattfeld, D., Kaminski, B., 2020. Binary driver-customer familiarity in service routing. European J. Oper. Res. 286 (2), 477–493.
Vela, C.R., Afsar, S., Palacios, J.J., González-Rodríguez, I., Puente, J., 2020. Evolutionary tabu search for flexible due-date satisfaction in fuzzy job shop scheduling. Comput. Oper. Res. 119, 104931.
Vidal, T., Laporte, G., Matl, P., 2020. A concise guide to existing and emerging vehicle routing problem variants. European J. Oper. Res. 286 (2), 401–416.
Wassan, N., Wassan, N., Nagy, G., Salhi, S., 2017. The multiple trip vehicle routing problem with backhauls: formulation and a two-level variable neighbourhood search. Comput. Oper. Res. 78, 454–467.
Xu, Z., Cai, Y., 2018. Variable neighborhood search for consistent vehicle routing problem. Expert Syst. Appl. 113, 66–76.
Zbib, H., Laporte, G., 2020. The commodity-split multi-compartment capacitated arc routing problem. Comput. Oper. Res. 122, 104994.
Zhen, L., Lv, W., Wang, K., Ma, C., Xu, Z., 2020. Consistent vehicle routing problem with simultaneous distribution and collection. J. Oper. Res. Soc. 71 (5), 813–830.