

A tabu search heuristic for the heterogeneous vehicle routing problem on a multigraph

David S.W. Lai^a, Ozgun Caliskan Demirag^{b,*}, Janny M.Y. Leung^c

^a Department of Information, Logistics and Innovation, VU University Amsterdam, The Netherlands

^b Black School of Business, Penn State Erie, The Behrend College, Erie, PA 16563, United States

^c Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong

ARTICLE INFO

Article history:

Received 16 September 2014

Received in revised form 20 October 2015

Accepted 6 December 2015

Available online 11 January 2016

Keywords:

HVRP

Multigraph

Vehicle routing

Heterogeneous fleet

Fixed sequence arc selection

Tabu search

ABSTRACT

We study a time-constrained heterogeneous vehicle routing problem on a multigraph where parallel arcs between pairs of vertices represent different travel options based on criteria such as time, cost, and distance. We formulate the problem as a mixed-integer linear programming model and develop a tabu search heuristic that efficiently addresses computational challenges due to parallel arcs. Numerical experiments show that the heuristic is highly effective and that freight operators can achieve advantages in cost and customer service by considering alternative paths, especially when route duration limits are restrictive and/or when vehicles of smaller capacity are dispatched to serve remote customers.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction and practical motivation

Vehicle routing problems (VRPs) arise frequently in the delivery and collection of items between a central depot and a number of customer locations. In its basic form, the VRP is concerned with determining a set of minimum-cost vehicle routes such that each route starts and ends at the depot, each customer is visited exactly once, and the total load on a vehicle does not exceed the capacity. As one of the most widely-studied combinatorial optimization problems, the VRP has been adapted to address a variety of practical considerations such as multiple trips by vehicles, time windows, multiple depots, and mixed fleet of vehicles.

Typically, the problem is studied on a complete undirected graph with vertices corresponding to the customer locations and arcs corresponding to the links between those locations. When an underlying road network is not complete, this representation can be obtained by computing the shortest path between possible origin–destination points on the original network. As mentioned by Garaix et al. (2010), the shortest path is generally computed based on a single attribute such as travel time, and this results in the alternative routes with different attributes (travel cost, distance, etc.) not being considered in the solution space. Garaix et al. (2010) have addressed these alternative routes, which often represent realistic trade-offs in distribution activities (e.g., travel time versus cost), by building a multigraph representation of the road network and showing the cost savings for an on-demand transportation problem. Baldacci et al. (2006) have previously introduced the multigraph structure into a multiple disposal facilities and multiple inventory locations rollon–rolloff vehicle routing problem. In this paper, we adopt the idea of alternative route consideration and study a time-constrained heterogeneous vehicle

* Corresponding author at: 5101 Jordan Road, Burke 259, Erie, PA 16563, United States. Tel.: +1 (814) 898 6717; fax: +1 (814) 898 6323.

E-mail addresses: david.lai@vu.nl (D.S.W. Lai), ozc1@psu.edu (O. Caliskan Demirag), janny@se.cuhk.edu.hk (J.M.Y. Leung).

routing problem on a multigraph. Similar to the work by [Garaix et al. \(2010\)](#), we illustrate the gains achieved by the consideration of alternative routes.

In particular, we study the following problem. There is a mixed fleet of vehicles positioned at a depot and a set of customer locations with fixed demand requirements. The vehicles are different in regard to the characteristics of capacity, availability, and cost. The goal is to find the least-cost vehicle routes, starting and ending at the depot, such that each customer is served by exactly one vehicle in a single visit, and the total demand on the vehicle does not exceed its capacity. In the literature, this problem is referred to as the mixed fleet or heterogeneous vehicle routing problem and generalizes the capacitated vehicle routing problem by introducing different vehicle types (e.g., [Gendreau et al. \(1999\)](#) and [Yaman \(2006\)](#)). In our work, we consider an upper limit on the route duration, and we study the problem on a multigraph where multiple parallel arcs between each pair of vertices correspond to the alternative paths connecting the two customer locations in the underlying road network. The paths are differentiated with respect to the multiple criteria under consideration, for example, time, cost, distance, and usage (wear and tear), and these properties are summarized as the arc attributes in the multigraph. Following the naming convention in [Baldacci et al. \(2006\)](#), we refer to this problem as the heterogeneous vehicle routing problem on a multigraph (HVRP-MG). We provide a flow-based mathematical formulation for the HVRP-MG and propose a tabu search heuristic for its solution. Tabu search is a widely-used heuristic method for VRPs and has been shown to provide near-optimal solutions efficiently for several related problems (Section 2). The presence of parallel arcs, however, renders simple tabu search operations such as insertion complicated, because, unlike the tabu search applications designed for simple graphs, arc selection decisions must be made to calculate the insertion costs on a multigraph. We develop a new procedure for estimating the insertion costs efficiently and integrate it into the search. Using instances with two attributes for the parallel arcs, our numerical experiments demonstrate the effectiveness of the proposed tabu search heuristic and provide managerial insights from sensitivity analyses. The proposed solution method can be extended to address multiple attributes for the parallel arcs. Below we provide a numerical example to motivate the use of multigraph structure in a VRP.

The key contribution of this research is that it addresses two important but not often studied areas of the VRP, namely, the fixed-fleet time-constrained heterogeneous vehicle routing problem and the vehicle routing problem with alternative paths. The heterogeneous vehicle routing problem in which a limited number of vehicles are available for delivery or collection activities (i.e., a fixed fleet) is a complex variant of the VRP and is closer to the distribution and transportation problems in practice ([Baldacci et al., 2008](#)). The presence of alternative paths further increases the practical relevance of the problem because it can address many situations in which the consideration of trade-offs such as minimum cost and faster service provides competitive advantages for the distributors. The distinctive feature of our model is the integration of these two important variants of the VRP, which leads to some interesting managerial insights such as those relating to the cost savings attributable to the alternative paths. From the methodology point of view, we contribute by developing a tabu search heuristic that effectively deals with the computational challenges introduced by the parallel arcs. The arc selection heuristic that we integrate into the tabu search allows us to efficiently search the neighborhood of a given solution.

Example. Suppose that a set of customers, numbered from 1 to 18, are to be served by a fleet of 3 vehicles. For simplicity, we assume that all vehicles have 150 units of capacity. Suppose that there are two alternative paths of travel between every pair of customer locations and they are incorporated into the network by two parallel arcs (links), differentiated by the travel cost and travel time combination. One of these arcs represents the faster but more costly path and the other represents the slower but less costly path. The route duration is limited to 260 time units. The goal is to find the minimum-cost vehicle routes within the route duration limit. We note that the values of the customer demands, travel and dispatch costs, and travel and service times are chosen according to the base-case setting in the numerical analysis, which is presented later in the paper in Section 5. The details are not necessary to illustrate the main insights from this example and hence are omitted here for brevity.

We analyze two cases. In the first case, we ignore the parallel arcs and solve the problem on a simple graph by considering the less costly arcs only. In the second case, we solve the problem on a multigraph by considering both types of arcs. [Fig. 1](#) depicts the corresponding solutions and [Table 1](#) summarizes the information on the vehicle routes.

In the simple graph, vehicle 3 returns to the depot after visiting customer 1, but in the multigraph, the route is extended to include customers 3 and 13 by using the arcs $\{(1,13), (13,3), (3,0)\}$ that are all of faster but more costly types. These arcs facilitate route extension without violating the route duration limit, which would have been infeasible in the simple graph. Consequently, the assignment and sequence of customers in the other vehicles' routes are changed, and while the travel cost of vehicle 3 increases, the total transportation cost of all routes is reduced from 657.92 to 652.11. Therefore, savings in travel costs, e.g., fuel costs and toll charges, can be generated by taking advantage of the flexibility offered by the parallel arcs. For larger problems, it is also possible to realize savings in vehicle dispatch costs because service can be delivered with a smaller number of vehicles.

The practical motivation for this study comes from the distribution of fast-moving consumer goods, specifically, malt beverages and beer in China. In this industry, producers often rely on a highly fragmented, complex, and multi-tiered supply chain to reach their customers. (See [Fig. 2](#) for an illustration based on [Dai and Zhou \(2008\)](#).) One unique feature of these systems compared to their counterparts in other countries such as the United States is the abundance of agents in the middle tiers, especially the wholesalers. The wholesalers satisfy demand requests coming from multiple customer sources including the retailers. They are served by a number of distributors, who replenish their stocks from the producer. In a beer supply

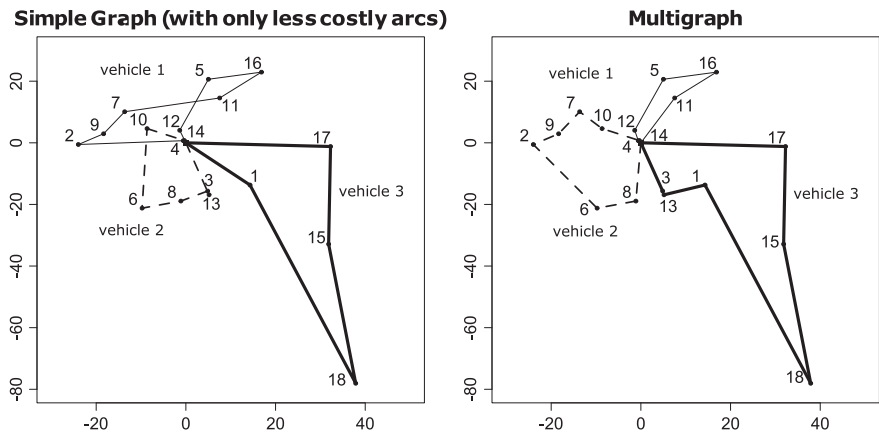


Fig. 1. Effects of considering alternative routes. (Left) Simple graph with less costly arcs only, (Right) Multigraph.

Table 1
Vehicle routes.

	Vehicle	Travel time	Vehicle load	Travel cost	Route
Simple graph ^a	1	170.00	120	199.92	(4, 2, 9, 7, 11, 16, 5, 12)
	2	121.47	147	163.86	(14, 10, 6, 8, 3, 13)
	3	258.01	92	294.14	(17, 15, 18, 1)
Multigraph	1	107.79	88	162.85	(4, 12, 5, 16, 11)
	2	148.77	135	186.23	(14, 10, 7, 9, 2, 6, 8)
	3	259.82	136	303.03	(17, 15, 18, 1, 13, 3)

^a Considering only the less costly arcs between pairs of customer locations.

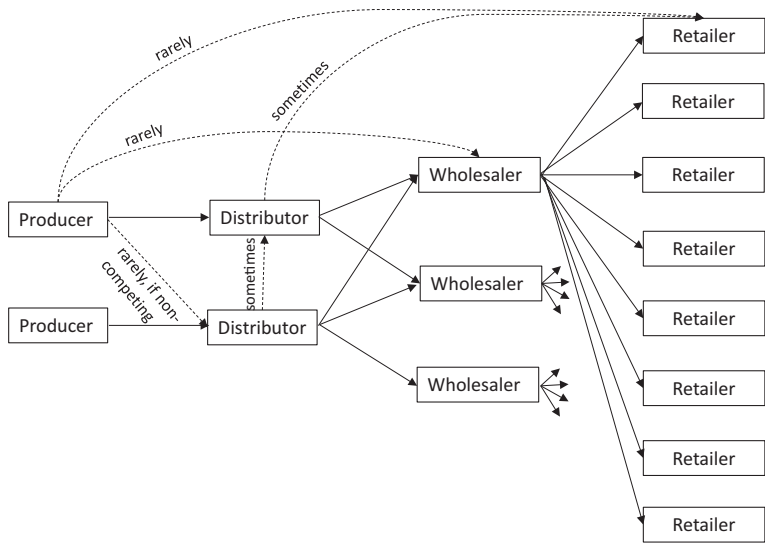


Fig. 2. Beer distribution network in China.

chain, while there may be around five distributors carrying one producer's brand (mostly) exclusively, it is not uncommon to have more than one hundred independent wholesalers who may sell goods from multiple brands. Major transportation activities take place between the producer and the distributors via long-haul carriers, and between the wholesaler or distributor and the customers in the form of local deliveries. Firms may use specialized services of 3PL companies for their distribution needs, although operating privately owned trucks or renting trucking services is more widespread in the industry. The sheer size of the network leads to frequent movements of the goods from upstream to downstream parties, which in turn contributes to increased transportation costs.

The transportation cost structure in China is shaped by the current logistics infrastructure and regulations that present some unique properties. For example, as reported by Dai and Zhou (2008), highway toll charges account for as much as 20–30% of the total logistic costs in China, whereas they are often assumed negligible compared to the driver wages, shipping, and handling costs in the United States. The carriers have the option of incurring lower toll charges by utilizing free public roads, but it comes at the expense of a slower and less reliable service. Other differences arise from fleet heterogeneity, which is usually more significant in the Chinese systems, and road access limitations for different vehicle types due to high population densities in Chinese cities. Furthermore, retailers often require frequent deliveries from their wholesaler, mainly because of space limitations in the stores and/or their reluctance to carry high stock levels given the customers' small shopping volumes caused in part by the relatively low levels of vehicle-ownership in Chinese households. As a result, wholesalers, who are involved in intense competition, often have to deal with deliveries of small quantities and unplanned vehicle routes while also considering the trade-offs between delivery costs and ability to provide service in short notice. (Other examples of academic research inspired by logistics problems in China include Chen et al. (2001), Fisher et al. (1986), Ma et al. (2012), and Yu and Qi (2014).)

These unique properties call for development of effective operational strategies to address alternative paths of travel (e.g., trading off cost and time), heterogeneous vehicles, and time-restricted delivery trips. Focusing on the shipment of products from a wholesaler to the retailers, we capture these characteristics by modeling a VRP on a multigraph and by considering multiple types of vehicles under a route duration constraint.

The rest of this paper is organized as follows. In Section 2, we review the related literature. Section 3 describes the problem and provides a mixed-integer linear programming formulation for the HVRP-MG. Solution methodology is discussed in Section 4, with a focus on the tabu search algorithm. In Section 5, we conduct a computational analysis and evaluate the performance of the tabu search heuristic as compared to a mixed-integer linear programming approach and two less sophisticated insertion-based heuristics. We also perform a sensitivity analysis to derive some managerial insights on the delivery operations in the presence of parallel arcs. Section 6 concludes the paper with a discussion of research limitations and extensions.

2. Literature review

Starting with the early work of Dantzig and Ramser (1959) and Clarke and Wright (1964), the classical VRP has been extensively studied in the operations research literature. Several books and articles survey this rich literature and provide information on the basic VRP and the solution methods (e.g., Golden et al. (2008), Toth and Vigo (2002), Laporte (2009), and Vidal et al. (2013)). Solution approaches to the VRP concentrate on exact methods, construction heuristics, local-improvement methods, and meta-heuristics, which are reviewed in the papers by Cordeau et al. (2002), Cordeau et al. (2005), Gendreau et al. (2002, 2007), Laporte (1992, 2007), and Potvin (2009). Likewise, the literature on the variants of VRP is quite well developed, studying a wide array of practical extensions such as time windows, (e.g., Desrochers et al. (1988) and Ollie and Gendreau (2005)), multiple trips by vehicles (e.g., Brandão and Mercer (1997) and Taillard et al. (1996)), multiple depots (e.g., Laporte et al. (1988) and Renaud et al. (1996)), and multiple types of vehicles (e.g., Gendreau et al. (1999) and Golden et al. (1984)).

The research stream on VRPs with multiple types of vehicles is the most relevant to our study. These problems are referred to as heterogeneous or mixed fleet VRPs and differ from the classical VRP in that a mixed fleet of vehicles with distinct capacities, fixed operating costs, and variable costs is used to serve the customers. Two variants of the problem arise with respect to the fleet size: The problem in which there is a limit on the number of vehicles available for each type (i.e., fixed fleet) is referred to as the Heterogeneous VRP (HVRP) and that with unlimited number of vehicles is referred to as the Fleet Size and Mix VRP (FSM) or the Vehicle Fleet Mix Problem (VFMP). Baldacci et al. (2008, 2010) review the related literature and give an overview of the heuristic and exact solution algorithms for both problems. Some mathematical formulations are provided by Choi and Tcha (2007), Gheysens et al. (1984), Golden et al. (1984), Salhi and Rand (1993), and Yaman (2006). Among these, Choi and Tcha (2007), Golden et al. (1984), and Yaman (2006) focus on exact algorithms for the VFMP and develop lower bounds. Heuristic approaches to the VFMP and its variants are studied in papers such as Brandão (2011), Desrochers and Verhoog (1991), Gendreau et al. (1999), Leung et al. (2013), Salhi and Osman (1996), Renaud and Boctor (2002), Wassan and Osman (2002), and Taillard (1999).

For the HVRP, the main solution approaches have been of heuristic type, with the exception of the works by Baldacci and Mingozzi (2009) and Pessoa et al. (2009). Specifically, construction heuristics and meta-heuristics have been applied to the HVRP and its variants (e.g., Taillard (1999), Tarantilis et al. (2003, 2004)). In general, meta-heuristics such as tabu search or large scale neighborhood search are reported to have superior performances (Baldacci et al., 2008). Recent research in this area has therefore focused on meta-heuristics, which often combine multiple solution methods by developing hybrid algorithms and includes Duhamel et al. (2012), Li et al. (2007, 2010), Liu et al. (2009), Penna et al. (2013), Prins (2009), and Subramanian et al. (2012). All of these papers address an HVRP with fixed costs and vehicle-dependent routing costs, which is also the main problem in our paper. Additionally, we consider two features: Route duration constraints and parallel arcs between pairs of vertices. While the route duration constraint is studied by Li et al. (2010), none of the papers in this stream consider parallel arcs. These arcs represent alternative paths of travel from a customer location to another in the underlying road network, and they provide more flexibility in constructing routes based on the different attributes of the arcs (e.g., more

costly but faster connections can be selected when delivery time limit is restricted.) We propose a mixed-integer linear programming formulation for the HVRP that can address multiple parallel arcs between pairs of vertices. Similar to Golden et al. (1984) and Yaman (2006), we extend the Miller–Tucker–Zemlin (MTZ) inequalities for the TSP (Miller et al., 1960) to model the capacity and subtour elimination constraints in our formulation; but we also incorporate the multigraph structure. In view of the findings reported by Baldacci et al. (2008), we focus on heuristic approaches and propose a tabu search heuristic that leads to near-optimal solutions in our numerical tests on parallel arcs with two attributes.

Another area of research that is closely related to our work deals with VRPs on a multigraph. Baldacci et al. (2006) introduce this structure to study a multiple disposal facilities and multiple inventory locations rollon–rolloff vehicle routing problem. In particular, they show that the problem can be modeled as a single depot time-constrained VRP on a directed multigraph. Assuming that the vehicles are identical and have unlimited capacity, they provide a set partitioning formulation of the problem and propose an iterative exact method. Our work differs from Baldacci et al. (2006) in that we incorporate a mixed fleet of vehicles and consider capacity limits in addition to the duration constraints on the vehicle routes. Furthermore, we formulate the problem using flow variables that indicate the vehicles' travel between customers, and provide a heuristic solution approach based on tabu search. In a different study, Garaix et al. (2010) incorporate alternative paths into vehicle routing problems by building a multigraph representation of the underlying road network. As discussed by these authors, some difficulties arise when parallel arcs are present between each pair of vertices in the graph. More specifically, although determining the exact schedule for a vehicle route is trivial in a simple graph after deciding the assignment of customers to the vehicles and the visiting sequence in the routes, this is no longer true for a multigraph structure. Due to the parallel arcs between the vertices, additional decisions must be made regarding the specific arc selection. Garaix et al. (2010) propose a dynamic programming algorithm for arc selection in the context of a dial-a-ride problem. They examine how the multigraph structure affects insertion operations and branch-and-price methods in solving these problems. Similar to Garaix et al. (2010), we consider multigraph structure in our model but our application area and solution approach differ from theirs as we study a heterogeneous vehicle routing problem and propose a tabu search heuristic with an efficient arc selection procedure. We demonstrate the effectiveness of our heuristic method by comparing it to their dynamic programming approach to arc selection.

We consider time limits on the vehicle routes, therefore, the literature on time (or distance) constrained VRPs is also relevant. Applications of these problems are provided in the papers by Assad (1988) and Laporte et al. (1984). With the objective of minimizing the total distance traveled, Laporte et al. (1984) and Laporte et al. (1985) provide formulations of the problem and develop exact solution methods based on constraint relaxation. Considering the minimization of total distance and number of vehicles used, Li et al. (1992) show that the optimal solutions under these objectives are closely related. We focus on minimizing the sum of travel and fixed dispatch costs while addressing time restrictions for the routes as an additional constraint. Different from these papers, we model multiple types of vehicles, which necessitates a heuristic solution approach rather than exact methods for realistic-sized problems.

First proposed as a local search method for combinatorial optimization problems (Glover, 1986, 1989), tabu search has been widely applied to VRPs with great success (e.g., Barbarosoglu and Ozgur (1999), Gendreau et al. (1994), Osman (1993), Rochat and Taillard (1995), Taillard (1993), Toth and Vigo (1998), and Xu and Kelly (1996)). The basic concept of tabu search is to explore the solution space iteratively by moving from one solution to the best neighboring solution that is not in a tabu list. The tabu list is maintained to avoid cycling, where recently examined solutions are not considered for a number of iterations unless they satisfy some aspiration criterion. The search may be improved by implementing intensification and diversification schemes that prevent the search being restricted to a limited portion of the search space and help explore the promising solutions more closely. Surveys on tabu search and other meta-heuristics for VRPs can be found in Cordeau et al. (2005) and Gendreau et al. (2002). Tabu search implementations for the classical VRPs are extended to incorporate multiple types of vehicles in the context of HVRP (e.g., Brandão (2011), Gendreau et al. (1999), Salhi and Osman (1996), Wassen and Osman (2002)). We propose an implementation that considers the multigraph structure, which has not been addressed previously. While our tabu search implementation maintains some of the attributes proposed by these papers, e.g., allowing the search to move to infeasible solutions and penalizing solutions with frequently moved vertices (Gendreau et al., 1999; Ho and Gendreau, 2006), it is adapted to handle the structure of parallel arcs between vertices through an efficient arc selection procedure that is incorporated in the search.

3. Problem formulation

In this section, we present a mixed-integer linear programming formulation for the HVRP-MG based on flow variables. Let $G(V, E)$ be a directed multigraph where V is a set of vertices and E is a set of arcs. Vertex $v_0 \in V$ denotes a depot from which all vehicles are operated, and the remaining vertices represent n customers. Each customer $i \in V \setminus \{v_0\}$ requires a certain number of units to be delivered to its location, representing the demand $d_i \in \mathbb{Z}^+$. Associated with this delivery is a service time denoted with $s_i \in \mathbb{R}^+$. The set E may contain parallel arcs between each pair of vertices that correspond to the alternative paths connecting the two locations in the underlying road network (Garaix et al., 2010). There is a heterogeneous fleet of vehicles with distinct capacities, fixed operating costs, and travel costs. The fleet is categorized into different types of vehicles, indexed over the set \mathcal{K} , so that vehicles of the same type are identical. For each type $k \in \mathcal{K}$, let $Q_k \in \mathbb{Z}^+$ denote the vehicle capacity, $f_k \in \mathbb{R}^+$ denote the fixed dispatch cost, and $m_k \in \mathbb{Z}^+$ denote the number of vehicles available. The travel time on an

arc $e \in E$ is given by $t_e \in \mathbb{R}^+$, and when a vehicle of type k travels through arc e , a travel cost $c_e^k \in \mathbb{R}^+$ is incurred. The objective is to determine a set of vehicle routes with the minimum total cost, subject to the following requirements:

- (i) Each route starts and ends at the depot and is associated with one vehicle type.
- (ii) Each customer is visited once by exactly one vehicle.
- (iii) The total demand served on a route of type- k vehicle does not exceed the vehicle capacity, Q_k .
- (iv) All vehicles return to the depot within a given time limit, L .
- (v) The number of type- k vehicles in use does not exceed the number of vehicles available, m_k .

All problem parameters are assumed to be known with certainty. Consistent with [Garaix et al. \(2010\)](#), the arcs in the multigraph represent Pareto-optimal road paths only. For instance, if each arc has two attributes, travel time and cost, then this implies that no arc exists for road paths that are dominated with respect to these two criteria; i.e., any path with a longer travel time and a higher travel cost can be ignored.

We next proceed to the formulation. In the following, we define the decision variables, $x_e^k \in \{0, 1\}$, $y_{ij} \in \mathbb{R}^+$, and $w_{ij} \in \mathbb{R}^+$. Let

$$x_e^k = \begin{cases} 1, & \text{if a vehicle of type } k \text{ travels on arc } e; \\ 0, & \text{otherwise;} \end{cases} \quad e \in E, k \in \mathcal{K}.$$

For all $i, j \in V$ with $i \neq j$, let $y_{ij} \in \mathbb{R}^+$ denote the total (cumulative) demand delivered when the vehicle leaves customer i to serve customer j . Similarly, let $w_{ij} \in \mathbb{R}^+$ denote the cumulative sum of service and travel times when the vehicle leaves customer i to serve customer j . If there is no vehicle traveling from customer i to customer j , both y_{ij} and w_{ij} are set to zero. For notational simplicity, let $E_{ij} \subset E$ denote the set of arcs from vertex i to vertex j , $\delta^+(i) \subset E$ denote the set of arcs that leave vertex i , and $\delta^-(i) \subset E$ denote the set of arcs that are incident to vertex i . Furthermore, it is convenient to treat the depot as a vertex with zero demand and zero service time.

Then, the HVRP-MG can be formulated as follows:

$$\min \sum_{k \in \mathcal{K}} f_k \sum_{e \in \delta^+(v_0)} x_e^k + \sum_{k \in \mathcal{K}} \sum_{e \in E} c_e^k x_e^k, \quad (1)$$

$$\text{s.t. : } \sum_{k \in \mathcal{K}} \sum_{e \in \delta^+(i)} x_e^k = 1, \forall i \in V \setminus \{v_0\}, \quad (2)$$

$$\sum_{e \in \delta^+(i)} x_e^k - \sum_{e \in \delta^-(i)} x_e^k = 0, \forall k \in \mathcal{K}, i \in V, \quad (3)$$

$$\sum_{e \in \delta^+(v_0)} x_e^k \leq m_k, \forall k \in \mathcal{K}, \quad (4)$$

$$\sum_{j \in V: j \neq i} y_{ij} - \sum_{j \in V: j \neq i} y_{ji} = d_i, \forall i \in V \setminus \{v_0\}, \quad (5)$$

$$y_{ij} \leq \sum_{k \in \mathcal{K}} \sum_{e \in E_{ij}} (Q_k - d_j) x_e^k, \forall i, j \in V : i \neq j, \quad (6)$$

$$\sum_{j \in V \setminus \{v_0\}} y_{v_0j} = 0, \quad (7)$$

$$\sum_{j \in V: j \neq i} w_{ij} - \sum_{j \in V: j \neq i} w_{ji} = s_i + \sum_{k \in \mathcal{K}} \sum_{e \in \delta^-(i)} t_e x_e^k, \forall i \in V \setminus \{v_0\}, \quad (8)$$

$$w_{ij} \leq \sum_{k \in \mathcal{K}} \sum_{e \in E_{ij}} (L - s_j - t_e) x_e^k, \forall i, j \in V : i \neq j, \quad (9)$$

$$\sum_{j \in V \setminus \{v_0\}} w_{v_0j} = 0, \quad (10)$$

$$w_{ij} \in \mathbb{R}^+, \forall i, j \in V : i \neq j, \quad (11)$$

$$y_{ij} \in \mathbb{R}^+, \forall i, j \in V : i \neq j, \quad (12)$$

$$x_e^k \in \{0, 1\}, \forall e \in E, k \in \mathcal{K}. \quad (13)$$

There are $|E||\mathcal{K}|$ binary variables and $2|V|^2$ non-negative continuous variables. The objective is to minimize the total fixed costs and travel costs. The set of constraints in (2) ensures that each customer is serviced by exactly one vehicle on a single delivery, that is, the demand is not split. Constraints in (3) and (4), respectively, balance the number of vehicles entering and leaving a vertex, and limit the number of vehicles in use. Constraints (5)–(7) ensure that all vehicle routes satisfy the capacity constraints and that the demand delivered when leaving the depot is zero. Finally, constraints (8)–(10) ensure that all vehicle routes satisfy the duration constraints and that the loading time at the depot is zero. As noted by [Yaman \(2006\)](#), no subtours will appear in any vehicle route due to the capacity (or duration) constraints. Notice that if (4) is removed, the model would determine the optimal fleet size for each vehicle type simultaneously.

Yaman (2006) presents a number of mixed-integer linear programming formulations for the HVRP. Our model is most similar to the formulation with disaggregated flow variables in the sense that the duration and capacity constraints are handled using variables associated with the arcs and vehicle types. As also mentioned by Yaman (2006), one advantage of disaggregating the flow variables by vehicle type is the ease in treating different variable costs, which is an important aspect in our setting. In addition, we incorporate the multigraph structure into the formulation. An HVRP model with similar definitions of flow variables y_{ij} and binary variables x_e^k is given by Baldacci et al. (2008), but there is no consideration of service duration constraints or parallel arcs in their formulation.

4. Solution methodology

Since the HVRP-MG has been formulated as a mixed-integer linear programming problem (Section 3), commercial optimization software such as CPLEX can be used to obtain the optimal solution for small-sized instances. For instances of practical scale, the problem formulation typically becomes quite large and such tools cannot obtain optimal solutions within a reasonable amount of computation time. Therefore, we propose a heuristic-based approach for solving the HVRP-MG. In particular, we develop a tabu search heuristic that has been proven successful for a wide variety of vehicle routing problems. This section presents the description of the tabu search heuristic and two other simpler heuristic methods against which we benchmark the performance of the tabu search heuristic.

4.1. Tabu search heuristic

Tabu search is a local search method that begins with an initial solution and explores the solution space by iteratively examining the neighboring solutions that are found by simple local modifications to the current solution. To avoid getting stuck in a poor local optimum, the search moves to the best neighboring solution even if this move results in deterioration of the objective function. Recently visited solutions are forbidden for a number of iterations, i.e., they are placed in a tabu list, in order to prevent cycling. Additional features developed for tabu search can be applied to improve the search, see, for example, Rochat and Taillard (1995), Gendreau et al. (1999), Taillard (1999), and Ho and Gendreau (2006).

Basic components of the tabu search heuristic include neighborhood structure, initial solution, tabu moves, aspiration criterion, and diversification/intensification mechanisms. We next describe how these components are designed to solve the HVRP-MG in our study. For better illustration, our description focuses on the case in which the multiple parallel arcs in between the vertices are differentiated with respect to two attributes, travel time and travel cost, where faster links represent more costly travel options and slower ones represent cheaper options. As we explain in the following sections, the solution methodology can be extended to address arcs with multiple attributes. When performing elementary tabu search operations such as insertion on a multigraph, the specific arc selection needs to be considered to evaluate a customer's insertion into a given route. Due to the time requirements associated with this process, we develop two variants of the tabu search algorithm depending on whether the insertion costs are computed by using an exact dynamic programming approach or an efficient heuristic approach for the arc selection. We describe the tabu search components based on the heuristic arc selection approach and present the variant with the exact arc selection approach as a benchmark case in Section 4.2.

4.1.1. Penalized objective function

Following Gendreau et al. (1994, 1999), Ho and Gendreau (2006) and others, we allow infeasible solutions in the search space for the tabu search. This idea is usually implemented by relaxing some of the constraints and incorporating them into the objective function with the use of self-adjusting penalty parameters, hence the name *penalized objective function*.

Let \mathcal{X} denote the set of solutions that satisfy requirements (i) and (ii) in Section 3, that is, every route starts and ends at the depot, and every customer is visited once by exactly one vehicle. For a solution $x \in \mathcal{X}$, let $\mathcal{R}(x)$ denote the vehicle routes that contain at least one customer, and for a vehicle route $p \in \mathcal{R}(x)$, let $V(p)$ and $E(p)$ be the vertices and arcs in the route. If a vehicle of type $k \in \mathcal{K}$ is assigned to route p , the travel cost $c(p)$, overload $q(p)$ and overtime $t(p)$, representing the violation of capacity and duration constraints, respectively, can be written as: $c(p) = f_k + \sum_{e \in E(p)} c_e^k$, $q(p) = \left[\sum_{i \in V(p)} d_i - Q_k \right]^+$, and $t(p) = \left[\sum_{e \in E(p)} t_e + \sum_{i \in V(p)} s_i - L \right]^+$.

After incorporating the penalties for possible violations, the penalized objective function value of a solution $x \in \mathcal{X}$ is found by $z(x) = \sum_{r \in \mathcal{R}(x)} (c(r) + \alpha q(r) + \beta t(r))$, where $\alpha \in \mathbb{R}^+$ and $\beta \in \mathbb{R}^+$ are the penalty weights that are self-adjusting in the search, similar to the implementations in Cordeau et al. (2001), Ho and Gendreau (2006) and Xue et al. (2014).

4.1.2. Initial solution

The initial solution for the tabu search algorithm is created with an insertion-based heuristic, which builds several routes at the same time. It begins by assigning exactly one arbitrarily selected customer to each of the randomly selected l vehicles, where l is a randomly generated integer between 1 and the total number of vehicles available. Then, the remaining customers are considered one by one, following a randomized order. For each customer, all possible locations in all routes are evaluated for insertion, and the customer is subsequently inserted into a vehicle route at the position that minimizes the *insertion cost* (the incremental change in the penalized objective function value $z(\cdot)$). We estimate the insertion costs

by using an efficient heuristic algorithm that performs arc selection for a fixed sequence of vertices on a route, as described in Section 4.1.4. (When the tabu search uses the exact solution approach for the arc selection, insertion costs are calculated by using dynamic programming.) Insertion costs are frequently calculated when evaluating the neighboring solutions in the course of the search.

4.1.3. Neighborhood structure

The solutions in the neighborhood of a given solution $x \in \mathcal{X}$, denoted as $\mathcal{N}(x)$, are the solutions that can be obtained by applying a single-move operation to the current solution. The move operation involves relocating a customer from its current route to another route at the location that minimizes the insertion cost, which is calculated by performing an efficient arc selection procedure in the destination route. (For the tabu search variant with exact arc selection, dynamic programming is used to calculate the insertion costs.)

In each iteration of the search process, all possible move operations for all customers are evaluated and the best one is subsequently performed. The best move operation is the one that leads to the lowest penalized objective function value after incorporating a diversification scheme that is explained shortly. To prevent cycling, if a customer has been moved from route r to route s in a given iteration, then moving the same customer back to route r is forbidden, or declared tabu, for the following θ iterations, where θ is a user-controlled parameter. A tabu move is allowed only when the resulting solution is feasible and has an objective function value that is better than that of the current best feasible solution found by the search. This is referred to as the aspiration criterion and it is intended to prevent the search from stagnating.

For diversification purposes, moves that are performed frequently in the search are penalized. We use a mechanism similar to that described by Gendreau et al. (1999) and Ho and Gendreau (2006), which is based on a penalty function $\phi(\cdot)$ applied only to non-improving solutions. For a given solution $x \in \mathcal{X}$, $\phi(x) = \lambda c(x) \sqrt{n} \vartheta_{ir}$, where n is the number of customers, ϑ_{ir} counts the number of times customer i has been moved to route r so far in the search, and λ is a positive parameter that controls the intensity of diversification. Different from Ho and Gendreau (2006), we do not include the number of non-empty vehicles in the penalty function since it is reflected in the fixed cost terms of the objective function.

In summary, given the current solution $x \in \mathcal{X}$, the best solution in the neighborhood, $\bar{x} \in \mathcal{N}(x)$, is the solution that minimizes $z(\bar{x}) + \phi(\bar{x})$, where $\phi(\bar{x})$ applies only if $z(\bar{x}) \geq z(x)$, and \bar{x} is non-tabu, unless it satisfies the aspiration criterion.

4.1.4. Arc selection for a fixed sequence of vertices

In tabu search heuristics for the VRP, insertion is an elementary operation that is performed frequently during the search when building the neighboring solutions – a customer is inserted into a vehicle route at a position that minimizes the insertion cost. It is an easy operation for a simple graph, however, it is difficult in the presence of multiple arcs. This is true even when the sequence of vertices is fixed, because decisions must be made regarding which arc to choose between every two successive vertices. In this section, we develop a procedure that can efficiently perform arc selection for a fixed sequence of vertices.

To simplify our presentation, we make some slight changes of notation. We suppose that $\mathcal{V} = (v_0, v_1, \dots, v_l, v_{l+1})$ is the vertex sequence after a customer is inserted into the route of a vehicle type $k \in \mathcal{K}$, where v_0 and v_{l+1} denote the depot. (See Fig. 3 for an example.) For all $i \in \{0, 1, 2, \dots, l\}$, let \mathcal{E}_i be the set of arcs between vertices v_i and v_{i+1} , and define \mathcal{E} as $\bigcup_{i=0}^l \mathcal{E}_i$. Then, letting c_e be the travel cost and t_e be the travel time for each arc $e \in \mathcal{E}$ by the corresponding vehicle type k , the insertion operation is concerned with selecting a combination of arcs $\mathbb{E} \in \mathcal{E}_0 \times \dots \times \mathcal{E}_l$ minimizing $K_1 + \sum_{e \in \mathbb{E}} c_e + \beta [\sum_{e \in \mathbb{E}} t_e - K_2]^+$, where K_1 and K_2 are non-negative constants such that $K_1 = f_k + \alpha [\sum_{i \in \mathcal{V}} d_i - Q_k]^+$ and $K_2 = L - \sum_{i \in \mathcal{V}} s_i$.

Essentially, the problem is equivalent to a multiple choice knapsack problem which is \mathcal{NP} -hard (Hifi et al., 2004; Nauss, 1978). In the multiple choice knapsack problem, a set of items each with a certain weight are subdivided into $l + 1$ mutually exclusive classes denoted with \mathcal{E}_i , $i \in \{0, 1, 2, \dots, l\}$, and exactly one item must be taken from each class to minimize the total cost of the items selected, subject to the constraint on the total weight. In the vehicle routing context with multigraphs, Garaix et al. (2010) introduced the problem as the Fixed Sequence Arc Selection Problem (FSASP) and proposed an exact solution method based on dynamic programming. While the dynamic programming approach can determine the insertion costs accurately, it is rather time consuming to implement in tabu search since the \mathcal{NP} -hard subproblem needs to be solved frequently during the search. Therefore, we propose a heuristic for the FSASP that efficiently determines the arc selection for a fixed sequence of vertices. In Section 5, we test the performance of the tabu search by using both the dynamic programming and the heuristic approach for the FSASP.

Given a fixed sequence of vertices on a particular route, the FSAS (Fixed Sequence Arc Selection) procedure (presented below) runs in polynomial time and takes into consideration that, if the arc selection is not altered after having inserted a customer, the insertion cost may not reflect the actual value of the customer sequence and may mislead the search to an undesirable direction. Notice that this is never an issue for a simple graph. See Fig. 4 for an example where the insertion of customer 2 into a vehicle route (1, 3) leads to a different arc selection between vertex 0 and vertex 3 in the multigraph. No such alteration is applicable in the simple graph.

As noted, we focus on the case where the arcs have two attributes, namely travel time and travel cost (which are negatively related). We assume that none of the arcs between any two vertices are dominated by another such arc on both attributes; that is, the parallel arcs between two vertices all lie on the Pareto-optimal frontier. We consider the relative

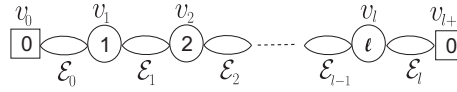


Fig. 3. Arc selection for a fixed sequence of vertices.

improvement of arc e compared to the current arc \bar{e}_i by considering the ratio $r(e) = \frac{c_e - c_{\bar{e}_i}}{t_{\bar{e}_i} - t_e}$. For arc e with higher cost ($c_e \geq c_{\bar{e}_i}$), this ratio represents the increase in cost per unit decrease in time, and is worthwhile if it is no bigger than β . If arc e has lower cost than \bar{e}_i , $r(e)$ denotes the reduction in cost per unit increase in time, and is worthwhile until the path exceeds the time limit, beyond which it is only worthwhile if $r(e) > \beta$.

Given the fixed vertex sequence on a route, $\mathcal{V} = (v_0, v_1, \dots, v_l, v_{l+1})$ and the set of arcs between the vertices, \mathcal{E}_i for $i \in \{0, 1, 2, \dots, l\}$, the FSAS procedure is as follows.

FSAS (Fixed Sequence Arc Selection) Procedure

- Step 1: Determine the longest-time path. If the duration constraint is satisfied, return the longest-time path (which has the smallest cost). Otherwise, go to Step 2.
- Step 2: Determine the shortest-time path. If the total duration of the shortest-time path is greater than or equal to the time limit, return the path with the following arcs: $e_i^* = \operatorname{argmin}_{e \in \mathcal{E}_i} c_e + \beta t_e$, $\forall i \in \{0, 1, \dots, l\}$. Otherwise, go to Step 3.
- Step 3: Start with the shortest-time path and then pick the arcs, one by one, following a predetermined order based on a ratio of cost to time difference, until the penalized objective function is no longer improving:
- Denote the currently selected arc in \mathcal{E}_i as \bar{e}_i . Let $\bar{\mathcal{E}} = \{\bar{e}_1, \bar{e}_2, \dots, \bar{e}_l\}$.
 - For all $i \in \{0, 1, \dots, l\}$ and $e \in \mathcal{E}_i \setminus \{\bar{e}_i\}$, determine the ratio of cost to time difference $r(e) = \frac{c_e - c_{\bar{e}_i}}{t_{\bar{e}_i} - t_e}$.
Sort arcs $e \in \mathcal{E} \setminus \bar{\mathcal{E}}$ with $c_e < c_{\bar{e}_i}$ in non-increasing order of $r(e)$. Let $\tilde{\mathcal{E}}_1$ be the sorted list.
Sort arcs $e \in \mathcal{E} \setminus \bar{\mathcal{E}}$ with $c_e \geq c_{\bar{e}_i}$ in non-decreasing order of $r(e)$. Let $\tilde{\mathcal{E}}_2$ be the sorted list.
 - Consider the arcs in $\tilde{\mathcal{E}}_1$ one by one in order. Let $\tilde{e}_j = \operatorname{argmax} r(e)$ be in \mathcal{E}_j . Replace \bar{e}_j by \tilde{e}_j , and remove all arcs in \mathcal{E}_j from $\tilde{\mathcal{E}}_1$, that is, let $\tilde{\mathcal{E}}_1 = \tilde{\mathcal{E}}_1 \setminus \mathcal{E}_j$.
 - As long as $z(\cdot)$ improves with the substitution (that is, when time limit is not exceeded or when $r(e) > \beta$), repeat from Step (c). Otherwise, go to Step (e).
 - Consider the arcs in $\tilde{\mathcal{E}}_2$ one by one in order. Let $\tilde{e}_j = \operatorname{argmin} r(e)$ be in \mathcal{E}_j . Replace \bar{e}_j by \tilde{e}_j , and remove all arcs in \mathcal{E}_j from $\tilde{\mathcal{E}}_2$, that is, let $\tilde{\mathcal{E}}_2 = \tilde{\mathcal{E}}_2 \setminus \mathcal{E}_j$.
 - If $z(\cdot)$ does not improve with the substitution, stop. Otherwise, repeat from Step (e).

The FSAS procedure determines the arc selection for a fixed sequence of vertices after inserting a customer on a particular route. Notice that the route returned from the FSAS procedure could be infeasible with respect to time and/or capacity restrictions since infeasible solutions are allowed in the search space. For a set of n vertices with at most p parallel arcs between each vertex pair, the FSAS algorithm is implemented with time complexity $O(pn \log(pn))$. We have presented the heuristic for the FSASP by focusing on the case with two attributes on the parallel arcs. When the arcs have multiple attributes, the subproblem that needs to be solved to determine the insertion costs in tabu search is the multiple dimensional, multiple-choice knapsack problem, which is \mathcal{NP} -hard (Hifi et al., 2004; Nauss, 1978). The FSAS procedure is extendable to objectives with multiple attributes by considering the ratio of the change in cost to the weighted sum of the changes in the non-cost attributes.

4.1.5. Diversification/intensification

We integrate two diversification devices into the tabu search. One is employed in the construction of the neighborhood solutions where the frequently performed operations are penalized (Section 4.1.3), and the other is used during the search

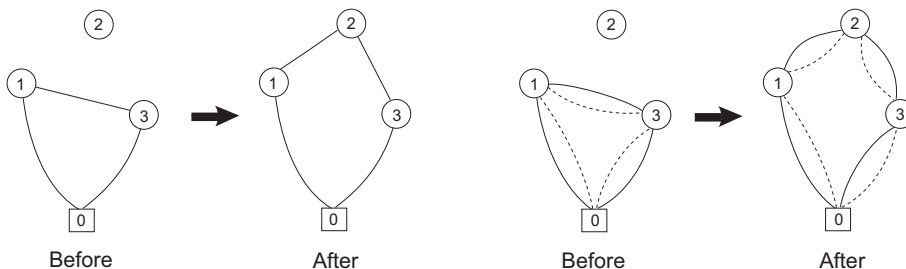


Fig. 4. (Left) Insertion operation in a simple graph and (right) in a multigraph.

process where the penalty weights for the violation of capacity and duration constraints are dynamically adjusted for better exploration of the search space. To allow high frequency of infeasible and non-improving solutions and hence encourage diversified structure, the penalty weights are initialized to small positive numbers. In particular, we follow the papers by Cordeau et al. (2001), Ho and Gendreau (2006), and Xue et al. (2014) and set the values of the penalty weights α and β to 1. The weights are adjusted during the search according to whether the solution in the next iteration satisfies the corresponding constraint: It is divided by $\delta + 1$ if there is no violation of the constraint. Otherwise, it is multiplied by $\delta + 1$, where $\delta \in \mathbb{R}^+$ is a user-defined parameter.

4.1.6. Search procedure

The search procedure consists of two phases. The first phase determines a good starting solution, and the second phase improves this solution by executing the main (tabu) search routine. The complete search procedure is shown in Algorithm 1 and the tabu search routine is presented in Algorithm 2.

The first phase of the procedure, given in lines 1–6 of Algorithm 1, starts by constructing an initial solution as described in Section 4.1.2. Next, this solution is improved by executing I iterations of the tabu search routine in Algorithm 2. The best solution visited during these iterations is recorded as a candidate for further improvement in the second phase. After repeating this process within a CPU time limit of ρ_1 seconds, the best solution is returned from the first phase, and the second phase starts. In the second phase (lines 7–10 of Algorithm 1), the best solution found in the first phase is improved by tabu search for ρ_2 CPU seconds. Unless stated otherwise, the search procedure stops when the total CPU time spent on both phases (ρ) reaches 60 s, and the best solution visited in the two phases is reported. The values for the user controlled algorithmic parameters in the search procedure, i.e., I , ρ , ρ_1 , ρ_2 , λ , δ and θ are given in Table 2.

Algorithm 1. Search Procedure

```

1: set  $z^* = \infty$  and  $x^*$  as an empty solution, and let  $\alpha^* = 1$  and  $\beta^* = 1$ .
2: do
3:   set  $x$  as an initial solution constructed by the process in Section 4.1.2 with  $\alpha = 1$  and  $\beta = 1$ .
4:   improve solution  $x$  using Tabu Search for  $I$  iterations, update  $\alpha$  and  $\beta$  in every iteration according to the procedure
     in Section 4.1.5, set  $\bar{x}$  as the returned solution, and  $\bar{\alpha}$  and  $\bar{\beta}$  as the updated penalties.
5:   if  $\bar{x}$  is feasible and  $c(\bar{x}) < z^*$  then set  $x^* = \bar{x}$ ,  $z^* = c(\bar{x})$ ,  $\alpha^* = \bar{\alpha}$  and  $\beta^* = \bar{\beta}$ .
6: while CPU time is less than  $\rho_1$  seconds
7: Set  $\alpha = \alpha^*$  and  $\beta = \beta^*$ .
8: improve solution  $\bar{x}$  using Tabu Search for  $\rho_2$  CPU seconds, and set  $\bar{x}'$  as the returned solution.
9: if  $\bar{x}'$  is feasible and  $c(\bar{x}') < z^*$  then set  $x^* = \bar{x}'$  and  $z^* = c(\bar{x}')$ .
10: return  $x^*$ .

```

Algorithm 2. Tabu Search

```

1: Data: initial solution  $x$ ; parameters  $\lambda$ ,  $\delta$ ,  $\theta$ ; penalty parameters  $\alpha$  and  $\beta$ ; and stopping condition.
2: if  $x$  is feasible, set  $z^* = c(x)$  and  $x^* = x$ ; otherwise, set  $z^* = \infty$  and  $x^* = x$ .
3: for all  $\bar{x} \in \mathcal{N}(x)$  do
4:   estimate (by using the FSAS procedure) the travel cost  $c(\bar{x})$ , overload  $q(\bar{x})$ , overtime  $t(\bar{x})$ , and diversification penalty
      $\phi(\bar{x})$ .
5: while stopping condition is not satisfied do
6:   select  $\bar{x} \in \mathcal{N}$  that
7:     – minimizes  $c(\bar{x}) + \alpha q(\bar{x}) + \beta t(\bar{x}) + \phi(\bar{x})$ 
8:     –  $\bar{x}$  is non-tabu or it satisfies the aspiration criteria
9:   determine  $c(\bar{x})$ ,  $q(\bar{x})$  and  $t(\bar{x})$  (by using Dynamic Programming)
10:  set the reverse move tabu for  $\theta$  iterations
11:  if  $\bar{x}$  is feasible and  $c(\bar{x}) < z^*$  then set  $x^* = \bar{x}$ ,  $z^* = c(\bar{x})$ , and perform intra-route improvement procedure on  $x^*$ .
12:  if  $\alpha$  and  $\beta$  have not been updated in the last  $n - 1$  iterations then
13:    if  $q(\bar{x}) = 0$  then set  $\alpha = \frac{\alpha}{1+\delta}$ ; else if  $\alpha \leq c(x^*)$  then set  $\alpha = \alpha + \alpha\delta$ .
14:    if  $t(\bar{x}) = 0$  then set  $\beta = \frac{\beta}{1+\delta}$ ; else if  $\beta \leq c(x^*)$  then set  $\beta = \beta + \beta\delta$ .
15:    Perform intra-route improvement procedure on  $\bar{x}$ .
16:  set  $x = \bar{x}$ 
17:  for all  $\bar{x} \in \mathcal{N}(x)$  do
18:    update  $c(\bar{x})$ ,  $q(\bar{x})$ ,  $t(\bar{x})$ ,  $\phi(\bar{x})$  (by using the FSAS procedure)
19: return  $x^*$ 

```

Table 2
Parameter values for the tabu search heuristic.

Parameter	Value
CPU time-limit (ρ)	60 s
CPU time spent in the first phase (ρ_1)	0.2ρ s
CPU time spent in the second phase (ρ_2)	0.8ρ s
Number of Tabu Search iterations in the first phase (I)	$\lceil 4n \rceil$
Diversification intensity (λ)	$0.000001 \sqrt{n}$
Penalty update factor (δ)	0.5
Tabu tenure (θ iterations)	$\lceil 5\log_{10}(n) \rceil$

Lines 3–4 of the tabu search routine (Algorithm 2) show the efficient procedure that we use in exploring the neighborhood of a solution. Before we relocate a customer from its current route to another route, we evaluate all such possible moves for all customers, but we do so by estimating the insertion costs using the FSAS procedure. Only after the best move is identified (lines 7–9), do we relocate the customer to the destination route and determine the best insertion position by solving the FSASP with dynamic programming. If a new best feasible solution is identified, an intra-route improvement procedure is performed on this solution where a customer is randomly picked and then reinserted into the best location of the same route, until no further improvement is possible. This intra-route improvement procedure is also performed after the penalty parameters are updated, which is done occasionally (every n iterations) to maintain accurate insertion costs to guide the search while balancing the computational requirements. Whenever the search moves to a neighboring solution, the insertion costs are again updated by using the FSAS procedure for efficiency (line 18). The computational time is manageable since we only need to update the insertions costs of the move operations that are involved with the modified routes.

As we show in the numerical experiments section, the mixed approach of using the heuristic FSAS procedure for determining the best move and the exact dynamic programming method for the optimal positioning of the customer has advantages in computational efficiency and is highly effective. This strategy is also consistent with a growing stream of research in neighborhood search-based heuristics. In the generic tabu search algorithms, all of the neighborhood solutions as defined in the neighborhood structure have to be evaluated in order to determine the search direction. However, in many cases, e.g., when the number of solutions in the neighborhood is very large or when evaluating a solution is time consuming, it might not be worthwhile to spend so much computational effort in one iteration of the tabu search. As the interest in incorporating side constraints to analyze rich variants of VRPs is increasing, such computational requirements represent a concern and hinder the application of the tabu search heuristic to many practical cases. On the other hand, methodologies such as large neighborhood search or very large scale neighborhood search often explore exponentially large and/or complex neighborhoods by use of heuristics in polynomial time (Pisinger and Ropke, 2010) or they restrict the neighborhood to a subset that can be searched efficiently. In our case, the number of neighborhood solutions for each iteration is only $O(n^2)$; however, it is time consuming to evaluate each of these solutions since it requires solving the \mathcal{NP} -hard FSASP several times. With this in mind, we use a polynomial time heuristic (i.e., the FSAS procedure) to examine all of the solutions in the neighborhood first, and then evaluate the solution picked by the heuristic using dynamic programming. This approach prevents us from spending computational efforts in evaluating neighborhood solutions that are more likely to lead to poor solutions.

We note that the search procedure given in Algorithms 1 and 2 can be adapted to account for more than two attributes. In such a case, the penalty factor for the violation of the corresponding constraint need to be incorporated into the penalized objective function, similar to the factors α and β .

4.2. Benchmark heuristics

The primary solution method that we propose for the HVRP-MG is the tabu search heuristic described in Section 4.1, where the insertion costs used to examine the neighboring solutions are calculated based on the FSAS procedure (Section 4.1.4). We refer to this method as TS-FSAS. Since the FSAS procedure is a heuristic approach, it is of interest to evaluate the quality of this approximation with regard to the exact dynamic programming approach. We address this by developing a variant of the tabu search heuristic where the insertion costs are determined using dynamic programming. In particular, lines 4 and 18 of Algorithm 2 are modified such that the corresponding costs are optimally determined by solving the resulting FSASPs with dynamic programming and line 9 is removed. We refer to this modified tabu search heuristic as TS-DP.

Additionally, we test the effectiveness of the tabu search heuristic compared to simpler heuristic methods that do not perform a sophisticated procedure to search for improved solutions. To this end, we use two benchmark heuristics, namely, an insertion heuristic (IH) and a greedy heuristic (GH), which differ according to the order in which the customers are considered to be assigned to a vehicle route. For comparison purposes, we develop two versions of these heuristic methods based on the calculation of the insertion costs, one based on the FSAS procedure and the other on dynamic programming. The resulting six heuristic solution approaches are summarized in Table 3. The details of the insertion and greedy heuristics are explained in the following.

Greedy heuristic: In this heuristic, customers are iteratively inserted into the vehicle routes and priority is given to those that incur the least change in the objective function value, i.e., the insertion cost. This is a simple heuristic that has been used

Table 3
Heuristic solution approaches.

Heuristic method	Insertion cost calculation	
	Exact (Dynamic Programming)	Heuristic (FSAS Procedure)
Greedy Heuristic	GH-DP	GH-FSAS
Insertion Heuristic	IH-DP	IH-FSAS
Tabu Search Heuristic	TS-DP	TS-FSAS

by Ropke and Pisinger (2006) to compare various insertion procedures in a large neighborhood search heuristic for a pickup and delivery problem. If we directly apply the heuristic to the HVRP-MG, customers would always be inserted into a non-empty vehicle (if any) due to the high dispatch cost. We avoid this bias in the initialization step of the algorithm. At the initialization, the customers and vehicles of different types are randomly ordered, and a customer is inserted to each of the first l vehicles where l is a randomly generated integer value between 1 and the total number of vehicles. Then, at each iteration, the “best” unassigned customer is inserted into the “best” vehicle route at the “best” location. This is done by comparing the minimum insertion cost of each unassigned customer, which is found by considering all routes and all locations on the routes, and selecting the one that leads to the lowest such cost. The insertion costs are determined by solving the FSASP, either by dynamic programming (GH-DP) or the FSAS heuristic procedure (GH-FSAS), and both of the penalty weights α and β are set to large positive numbers (i.e., 1,000,000 in our computational tests) so that feasible solutions can be encouraged. A single pass of the algorithm stops when all customers have been inserted, and the algorithm is rerun multiple times within the specified computational time limit where different randomized orders are used in initialization. The best solution resulting from these multiple passes is reported as the GH solution.

Insertion heuristic: In this heuristic, customers are selected at random and inserted into vehicle routes at a random order iteratively. The algorithm is initialized by following the procedure used in the GH. Next, the customers that are unassigned after the initialization are inserted one by one, following a random order, into a vehicle route at the position that minimizes the insertion cost. Similar to the GH, the insertion costs are calculated by either the dynamic programming approach (IH-DP) or the FSAS heuristic procedure (IH-FSAS), and α and β are set to large positive numbers to increase the likelihood of obtaining feasible solutions. We generate solutions with various randomized insertion orders and various number of customers at the initialization, and report the best solution found in the computational time limit as the IH solution.

5. Computational experiments

In this section, we test the performance of the tabu search heuristic and provide some managerial insights from sensitivity analyses. The computational experiments are performed on generated data that reflect the practical applications that have motivated our work and give us flexibility in constructing different scenarios for the sensitivity analysis. When selecting the values for the parameters in the experiments, we have benefited from the study by Dai and Zhou (2008), which provides detailed information and important statistics for the beer distribution industry in China.

5.1. Experimental setting

In our experiments, we focus on the instances where the number of parallel arcs between every pair of vertices is two and each of these arcs represents alternative routes that are differentiated with respect to two attributes, namely, travel time and travel cost. While our solution methodology can be extended to address more general cases with multiple parallel arcs and arc attributes, we believe that our experimental setting is sufficiently rich to study the main trade-offs involved in operating with alternative routes in practice.

Table 4 shows the values of the model parameters in the base case. The distribution of customers is intended to mimic a densely populated region close to the depot and a sparsely populated remote region. For simplicity, we assume that two types of vehicles are in use and those with larger capacity have higher dispatch and travel costs per unit distance due to higher fuel costs and toll charges. That is, we consider a correlated fleet in which the fixed and variable costs grow with the vehicle capacity (Prins, 2009). This is consistent with research on per-mile costs of vans (semi-trucks or pick-ups) and trucks (e.g., Barnes and Langworthy (2003)), as well as the industry practice (Dai and Zhou, 2008). The arcs have two attributes, travel cost and travel time, which we assume are symmetric; that is, for a given vehicle type, the travel cost and time going from vertex i to vertex j are the same as those going from vertex j to vertex i . With a slight change in the indexing, we denote the travel time and cost between i and j on vehicle k with t_{ij}^k and c_{ij}^k , respectively. We calculate travel times with the Manhattan distance metric, which helps us to reflect road network characteristics such as those that can be found in a city or town. There are two parallel arcs between every vertex pair, representing alternative routes with respect to travel time and cost characteristics. The travel time (t_{ij}^k) and cost (c_{ij}^k) of one of these arcs are generated according to the base-case values shown in Table 4. The values of the second arc, corresponding to the faster but more costly travel option, are generated

Table 4

Base-case parameter values for the numerical experiments.

Parameter	Description	Values
d_i	Demand of customer i	Uniform[5,35], integer
s_i	Service time at customer i	Uniform[1 + 0.2 d_i , 2 + 0.2 d_i]
$(r \cdot \cos\gamma, r \cdot \sin\gamma)$	Location coordinates	$r \sim \text{Uniform}[0,25]$ (80% of customers) $r \sim \text{Uniform}[25,100]$ (20% of customers) $\gamma \sim \text{Uniform}[0,2\pi]$
L	Route duration limit	260 time units
t_{ij}	Travel time	Manhattan distance between customer i and j
$ K $	Number of vehicle types	2 (Small and Large capacity vehicles)
m_k	Number of vehicles of type k	Small capacity vehicles: $\max\left(3, \left\lceil \frac{\sum_{i \in V \setminus \{v_0\}} d_i}{150} \right\rceil\right)$ Large capacity vehicles: $\max\left(3, \left\lceil \frac{\sum_{i \in V \setminus \{v_0\}} d_i}{300} \right\rceil\right)$
Q_k	Vehicle capacity	Small capacity vehicles: 150 Large capacity vehicles: 300
f_k	Fixed dispatch cost	Small capacity vehicles: Uniform[95,105] Large capacity vehicles: Uniform[145,155]
c_{ij}^k	Travel cost from customer i to j for vehicle type k	
	$c_{ij}^k = (\epsilon^k + \eta_{ij}^k)$	
	ϵ^k : Per unit distance fuel cost	Small capacity vehicles: Uniform[0.5, 1.1] Large capacity vehicles: Uniform[1.4, 2.0]
	η_{ij}^k : Per unit distance toll charge	Small capacity vehicles: Uniform[0.2, 0.3] (appear on 50% of the arcs) Large capacity vehicles: Uniform[0.4, 0.5] (appear on 50% of the arcs)

as follows: If a vehicle of type k travels from vertex i to vertex j at cost c_{ij}^k and time t_{ij} , then the second arc has cost $R_1 c_{ij}^k$ and time $R_2 t_{ij}$ where $R_1 \sim \text{Uniform}[1.1, 1.3]$ and $R_2 \sim \text{Uniform}[0.7, 0.9]$.

Algorithmic parameters are tuned by using one set of instances (training set), but all approaches are evaluated by using another set of instances (test set) and no parameter tuning is done on the test set to avoid over-fitting. When evaluating the performance of different solution approaches, the same set of instances are used for a fair comparison. All of the experiments have been conducted on a desktop personal computer running Windows 7 with an Intel Core i5-4570 processor with 3.2 GHz and 4 GB of main memory. Algorithms have been implemented in C++ and compiled using Visual Studio 2013. In some of the examples, we present the numerical results using Box-and-Whisker plots, where the median value is shown with a heavy solid line within the box that is composed of the upper and lower quartiles, and the outliers beyond the minimum and maximum values are shown with circles.

5.2. Performance of the solution approaches

We begin by testing the quality of the heuristic solutions against the solutions obtained from the mixed-integer programming model (MIP) by using CPLEX 12.6. CPLEX solves the problem to optimality in small-sized instances (i.e., $n \in \{14, 15, 16, 17\}$); however, for instances with 18 or more customers, it is difficult to obtain the optimal solutions within a reasonable amount of computation time. Therefore, we use the best solutions that CPLEX provides in 2 h of CPU time as the benchmark results (i.e., MIP results) and also report the lower bounds given by CPLEX. For the small-sized problems, we generate 5 instances at each number of customers and report the average cost from these instances. For the larger-sized problems, the number of customers is selected from the set $\{20, 25, 30\}$. We generate 5 instances for each case and report the results individually.

Table 5 summarizes the results.¹ (In this table and others, the result from the best performing method is displayed in bold.) For $n \in \{14, 15, 16, 17\}$, the solutions found by the MIP are optimal and the TS heuristic is able to produce the optimal solutions in almost all of the instances. As the number of customers increases, the performance of the MIP worsens due to the problem size, and CPLEX is not able to find the optimal solutions within the computational time limit. In these instances, the TS heuristic produces better solutions than the MIP. In the small-sized instances where the TS is able to find the optimal solution, the GH and IH do not perform well, and although they perform better in the middle range of problem sizes, their performance deteriorates as the problem size increases. Overall, the two variants of the TS heuristic outperform the other solution methods, and the TS heuristic with the FSAS procedure performs better than the TS heuristic with the DP approach.

We now consider instances with 100 customers and compare the performance of the heuristic methods under different computational time limits. For each CPU time limit, 25 instances are generated and the average cost from these instances is reported. Table 6 shows the results. The TS-FSAS heuristic has the best performance, while the GH-DP performs the worst in most of the instances. Typically, the IH and GH generate a large number of feasible solutions quickly, but the TS-FSAS heuristic is able to find better solutions within a few seconds. The TS-DP heuristic, on the other hand, does not perform as well as the FSAS variant, especially when the time limit is restrictive because of the high computational requirements.

¹ All of the instances and results of the computational experiments in this section are available for download at <<https://psu.app.box.com/hvrp-mg>>.

Table 5

Performance of the GH, IH, and the TS heuristic with respect to the MIP and MIP Lower Bound.

# of customers	Objective function value							
	MIP	MIP LB	GH-DP	GH-FSAS	IH-DP	IH-FSAS	TS-DP	TS-FSAS
14	538.69	538.69	638.17	638.17	638.17	638.44	538.69	538.69
15	493.30	493.30	577.50	576.48	576.48	576.48	493.62	493.62
16	526.33	526.33	526.33	526.33	526.33	528.81	527.38	526.33
17	650.18	650.18	650.41	650.30	650.30	650.18	650.18	650.18
20	818.39	795.82	832.78	821.89	821.89	821.89	818.39	818.39
20	775.86	754.41	777.71	776.21	776.21	777.72	773.30	773.30
20	799.49	769.96	802.39	801.21	801.21	802.80	799.49	799.49
20	798.27	798.03	803.65	798.91	798.91	798.91	798.27	798.27
20	791.52	766.82	799.79	794.46	799.01	793.86	791.52	791.52
25	1092.93	860.18	1015.48	1004.36	1004.36	999.75	998.56	994.04
25	1084.19	866.62	1014.22	997.90	1010.23	992.45	1028.37	992.45
25	1039.27	851.51	1032.79	996.00	996.00	998.71	987.82	987.82
25	1063.57	847.45	988.28	988.40	988.28	975.54	973.41	972.92
25	1071.02	844.05	1015.88	987.19	987.19	985.61	970.37	970.37
30	1080.25	840.48	1152.92	1086.13	1118.60	1073.54	1024.68	1012.99
30	1067.44	828.53	1115.41	1089.22	1077.00	1020.78	1011.45	989.32
30	1042.68	838.93	1023.33	1050.24	1023.33	1013.10	994.05	981.85
30	1052.63	815.15	1120.51	1080.85	1116.05	1054.22	1017.40	1015.60
30	1244.49	852.17	1111.78	1111.78	1081.34	1075.23	1027.72	1026.28
Average	738.99	667.74	759.11	752.60	753.03	747.25	716.12	713.39

Table 6

Performance under different computational time limits.

CPU time ^a	GH-DP	GH-FSAS	IH-DP	IH-FSAS	TS-DP	TS-FSAS
10.00	4296.51	3922.57	4007.82	3687.54	3095.86	2853.62
20.00	4212.59	3866.60	3965.96	3648.25	3016.89	2823.61
30.00	4143.74	3814.44	3932.83	3636.30	2971.29	2794.39
40.00	4135.80	3803.77	3911.03	3620.16	2945.89	2790.38
50.00	4116.99	3800.02	3908.93	3616.92	2940.54	2786.40
60.00	4103.43	3792.38	3903.75	3611.08	2921.82	2781.57
70.00	4095.91	3783.82	3894.85	3605.90	2916.56	2781.62
80.00	4091.23	3775.28	3894.19	3605.49	2891.26	2778.21
90.00	4091.23	3768.72	3886.93	3596.48	2872.64	2777.62
100.00	4083.15	3767.05	3886.93	3590.47	2863.78	2776.62
200.00	4060.87	3716.34	3845.42	3556.15	2846.04	2749.33

The result from the best performing method is displayed in bold.

^a CPU Time is in seconds.

Next, we compare the performance of the TS-FSAS and TS-DP heuristics for varying numbers of customers. For completeness, we also report the performance of the GH and the IH. We select the number of customers between 20 and 200, at the increments of 10. For each number of customers, 25 instances are generated and the average cost is reported (Table 7). The results show that the TS-FSAS and TS-DP heuristics outperform the two variants of the GH and the IH in all instances. Consistent with the observations in Tables 5 and 6, the GH-DP provides the worst performance. In general, the performance of the IH-FSAS is slightly better than the GH-FSAS, but both heuristics perform significantly worse than the TS heuristics, especially when the problem size increases, as shown by the percentage gap values. This suggests that the less-sophisticated solution methods, i.e., the GH and the IH, are not expected to perform well for the practical cases of the HVRP-MG and the development of a more complex meta-heuristic such as the proposed tabu search is justified. Moreover, we observe that the TS heuristic provides better quality solutions when the FSAS procedure is used for calculating the insertion costs instead of the DP approach. In general, the FSAS procedure is quite effective in guiding the search process in the right direction; furthermore, it is computationally more efficient in exploring the search space than DP. In view of these findings, we select the TS-FSAS as the main heuristic method to conduct further analysis in the next section.

Before we proceed, we evaluate the performance of the TS-FSAS heuristic by comparing its results with the best-known solutions on some benchmark instances. For this purpose, we take the problem data provided in Golden et al. (1984) and use the best-known solutions for the HVRPs as reported by Li et al. (2007) and Prins (2009). In these instances, the number of customers ranges from 50 to 100 and the number of vehicles ranges from 3 to 6, while the fixed costs and service times are assumed zero and no route duration limit exists. For consistency, we apply our algorithm by setting an arbitrarily large duration limit so that the time constraint is negligible. Since the prior works have focused on a simple graph structure, we slightly modify the algorithm given in Section 4.1.6, which has been developed to address parallel arcs. In particular, the FSAS procedure is not performed in the simple graph as the sequence of vertices will automatically determine the arcs in

Table 7

Performance on instances with different number of customers.

# of customers	GH-DP	GH-FSAS	IH-DP	IH-FSAS	TS-DP	TS-FSAS	GH-TS % ^A	IH-TS % ^A
20	797.14	794.78	794.19	794.01	791.09	791.06	0.37	0.47
30	1101.24	1076.12	1078.41	1037.15	1009.66	999.54	7.66	3.76
40	1854.60	1829.43	1818.85	1802.79	1501.75	1484.24	23.26	21.46
50	2117.90	1872.56	1953.23	1819.50	1753.89	1614.92	15.95	12.67
60	2618.93	2557.12	2549.36	2502.60	2155.45	2093.63	22.14	19.53
70	3045.93	3018.16	3002.45	2970.03	2208.89	2158.23	39.84	37.61
80	2915.64	2694.67	2755.90	2562.43	2320.97	2202.45	22.35	16.34
90	3391.24	3122.05	3215.24	2996.22	2607.93	2493.80	25.19	20.15
100	4103.43	3792.38	3903.75	3611.08	2921.82	2781.66	36.34	29.82
110	4925.46	4775.24	4761.19	4651.63	3982.58	3626.59	31.67	28.26
120	4087.37	3923.52	3901.34	3803.70	3257.07 ^a	3018.67	29.98	26.01
130	5619.72	5201.49	5378.12	4939.45	4275.41 ^b	3982.56	30.61	24.03
140	6213.68	5752.41	5931.30	5497.16	4429.71	4270.42	34.70	28.73
150	6744.98	6115.17	6230.54	5803.00	4863.13	4572.53	33.74	26.91
160	6964.63	6067.40	6261.40	5747.03	4451.34 ^c	4192.81	44.71	37.07
170	8872.86	8307.34	8353.57	7776.50	5818.34	5398.27	53.89	44.06
180	8368.47	7657.33	7738.29	7191.55	5578.78	5276.93	45.11	36.28
190	7685.72	7287.15	7285.60	6922.89	5564.34	5086.16	43.27	36.11
200	7926.44	7704.36	7629.90	7525.67	5268.93	4778.94	61.21	57.48

^a(^b)(^c): One (two) [two] instances with infeasible solutions are ignored in the average calculation.

The result from the best performing method is displayed in bold.

^A Percent gaps are based on the FSAS variants of the GH, IH and the TS heuristic.

a route. As can be seen from the results in Table 8, the solutions found by our tabu search heuristic are within 1.6% of the best-known results. Our focus in this paper is on developing a solution method for the time-constrained HVRP on a multi graph, i.e., one that addresses the parallel arc structure through procedures such as FSAS; therefore, we devote our further efforts to provide insights about the use and benefits of parallel arcs in forming vehicle routes.

5.3. Sensitivity analysis and managerial insights

We next perform sensitivity analyses to generate managerial insights. To address realistic settings, we consider instances with 100 customers under various test scenarios. We solve all instances by using the tabu search heuristic summarized in Algorithms 1 and 2, i.e., TS-FSAS, and select the values of the algorithmic parameters according to Table 2. In the base case, the problem parameters take the values shown in Table 4. In Section 5.3.1, we generate additional scenarios by varying the route duration limit, vehicle capacities, and the distribution of the customers. When the effect of one parameter is tested, other parameters are kept at their base-case values. In Section 5.3.2, we illustrate the gains due to the alternative routes, and in Section 5.3.3, we provide further insight on the effect of customer distribution.

5.3.1. Effects of route duration limit, vehicle capacities, and customer distribution on transportation costs

We test the values of L , the route duration limit, from 200 to 300 in increments of 10 units. The base-case values of the vehicle capacities are denoted with a ratio of 1, and the other scenarios are generated by varying this ratio within the set {0.6, 0.7, 0.8, 0.9, 1}, e.g., the ratio 0.6 implies 40% lower capacities than the base case. To obtain different distribution of customers, we vary the percentage of customers located in the remote region. This percentage is 20% in the base case, and the other values tested in the experiments are 35%, 50%, 65%, 80%. See Fig. 5 for an example illustration of the customer locations at different parameter values. For each specific combination of parameter values, we test 10 instances.

Fig. 6 shows the total cost of transportation including the dispatch and travel costs (i.e., the solution value) for the tested scenarios. The results illustrate the usual trade-off between transportation costs and customer service. As the route duration limit increases, it is possible to reduce transportation costs; however this comes at the expense of customer service, approximated by the total lead time to fulfill all customer orders. At the restricted values of the duration limit, the presence of parallel arcs helps firms to improve customer service by enabling the construction of routes within the specified duration limit. The figures also show that some cost savings are obtained due to economies of scale when the vehicles have larger capacities or due to reduced travel costs when only a small percentage of customers are located far from the depot. These characteristics are often difficult to control by an operating firm, therefore it is important that a balance is sought between the targeted lead time and transportation costs.

5.3.2. Gains due to the consideration of alternative routes

In practice, road networks present alternative ways (routes) of travel from one location to another with respect to attributes such as cost and time. When VRPs are studied on simple graphs, some of these attributes are omitted in constructing the graph since only one link is assumed between each pair of nodes. By incorporating alternative routes through a multi-graph structure, these attributes can be explicitly considered via multiple links between nodes and potentially generate

Table 8

Performance comparison with some benchmark instances on a simple graph.

Problem	Best-known solution	TS-FSAS	Gap%
13	1517.84	1538.16	1.34
14	607.53	615.516	1.31
15	1015.29	1020.27	0.49
16	1144.94	1146.59	0.14
17	1061.96	1080.29	1.73
18	1823.58	1846.71	1.27
19	1117.51	1142.79	2.26
20	1534.17	1588.92	3.57

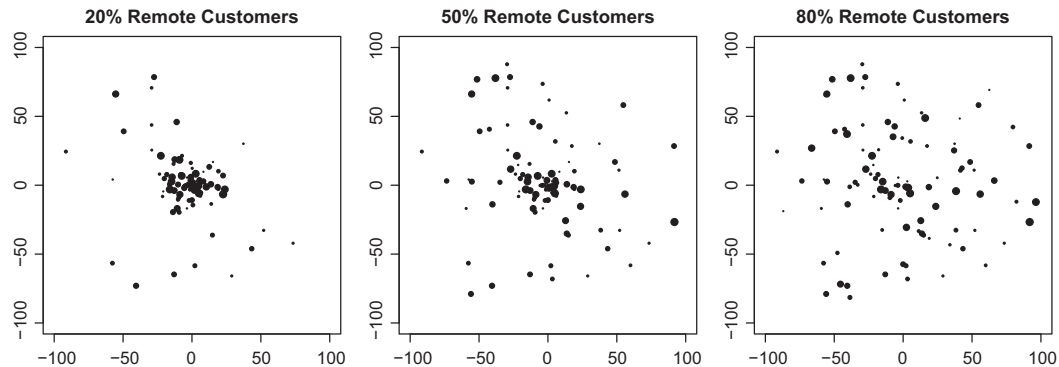


Fig. 5. Distribution of customers. A larger point represents a customer with a larger demand, and a percentage of 80% corresponds to approximately evenly distributed customers.

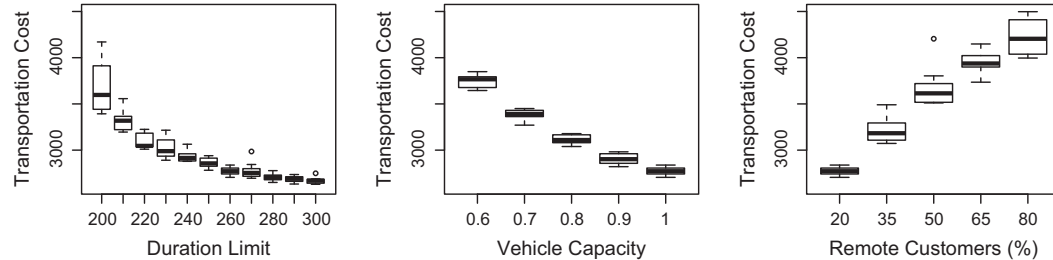


Fig. 6. The impact of the route duration limit, vehicle capacities and customer distribution on transportation costs.

some benefits such as cost savings and flexibility in distribution planning. [Garaix et al. \(2010\)](#) have quantified such benefits in an on-demand transportation problem. In this section, we investigate the potential benefits for an HVRP and identify circumstances under which significant gains can be generated by considering alternative routes.

[Fig. 7](#) illustrates the transportation cost savings when alternative routes are considered. We calculate the cost savings as the percentage change between the total transportation cost of a simple graph, containing only the set of less-costly and longer-duration arcs and that of a multigraph with both sets of arcs, i.e., $\frac{S-M}{S} 100\%$, where S and M respectively represents the total cost in the simple graph and multigraph.

The average amount of savings in transportation costs is around 5%, which is a significant value in distribution planning. The savings are more pronounced when the route duration limit is at lower values or when the vehicle capacity is large. When the time limit is more restrictive, the set of more-costly arcs that facilitate faster travel are selected more often and therefore their impact in generating savings is greater. At larger capacity values, a higher number of customers can be served in each trip and the vehicle routes can be further extended by utilizing the parallel arcs, which leads to larger savings mainly in dispatch costs. Furthermore, it is sometimes not only desirable but is necessary to consider alternative routes in distribution planning, especially when products have to be delivered within a short due date. For example, if we do not allow the parallel arcs and set the duration limit to a value less than 250, no feasible solution could be found in some of the instances. Parallel arcs enable the construction of vehicle trips within time limitations.

The savings are realized through two intertwined factors. One is the ability to construct more cost effective routes and the other is the ability to reduce the required fleet size. As we have seen in [Section 1](#), the more costly but faster arcs facilitate

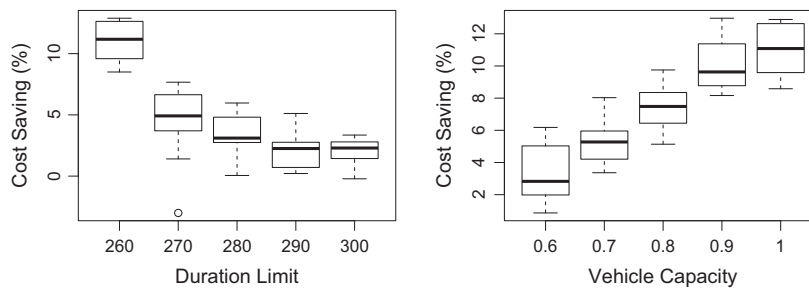


Fig. 7. Transportation cost savings due to parallel arcs.

extended routes and lead to reduced total traveling costs. In some cases, e.g., when the route duration limit is restricted, such arrangements can decrease the number of vehicle dispatches required and hence reduce the total fixed costs. We illustrate in Fig. 8 that it is possible to achieve savings from both of these factors, and the savings could be dominated by one of the two depending on the problem parameters. For example, as the fixed dispatch cost increases, more of the savings from parallel arcs are attributable to reduced dispatch costs.

Fig. 9 illustrates the utilization of parallel arcs in the routes of small and large capacity vehicles. We measure the utilization with a percentage, which is given by the ratio of the total number of more costly arcs traveled by a vehicle type on all trips to the total number of arcs traversed by the same vehicle type. The results show that the percentage of more costly arcs is higher in the routes of small capacity vehicles, implying that these vehicles tend to utilize the parallel arc structure more heavily. As the route duration limit becomes less restrictive, the percentage decreases steadily since the need for faster routes is reduced. This explains why large cost savings are observed for low duration limits, as we have seen in Figs. 7 and 8.

Figs. 7 and 9 show that the transportation cost savings are generated mainly by utilizing the more costly arcs in the routes of the small capacity vehicles and that the large capacity vehicles tend to avoid the more costly arcs. It is worthwhile to take a deeper look into the reasons behind this observation. To this end, we consider the customer distribution and the assignment of customers to vehicle types. Fig. 10 shows the percentage of remote customers served by the small capacity vehicles at different values of the route duration limit and vehicle capacity. The percentage is measured by the ratio of the number of remote customers visited by the small capacity vehicles to the total number of remote customers. In Fig. 11, we illustrate the time utilization in the routes of small and large capacity vehicles, where the utilization is measured by the ratio of total travel time to the route duration limit. We observe that most of the remote customers are served by the small capacity vehicles. Only when the duration limit is restrictive or the vehicle capacity is limited, a few customers in the remote region are served by the large capacity vehicles. At the same time, the small capacity vehicles tend to travel longer distances to serve the customers in the more remote regions, while the large capacity vehicles travel shorter distances to serve the customers nearby. Therefore, the trade-off between travel cost and time is more critical for the small capacity vehicles, and consequently, the consideration of alternative routes is more useful for this type of vehicles.

5.3.3. Further analysis on customer distribution

We perform further analysis on the impact of customer distribution over the network. By varying the percentage of customers located in the remote region, we can obtain different customer distributions, corresponding to, for example, uniformly or largely densely (sparsely) populated networks.

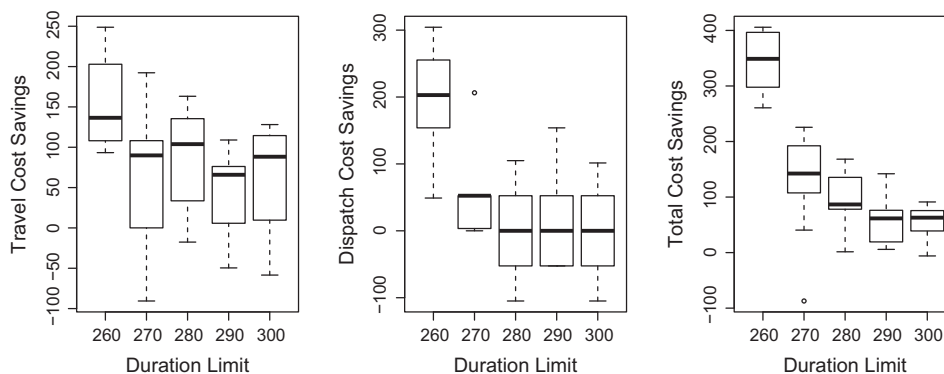


Fig. 8. Savings due to reduced traveling and dispatch costs.

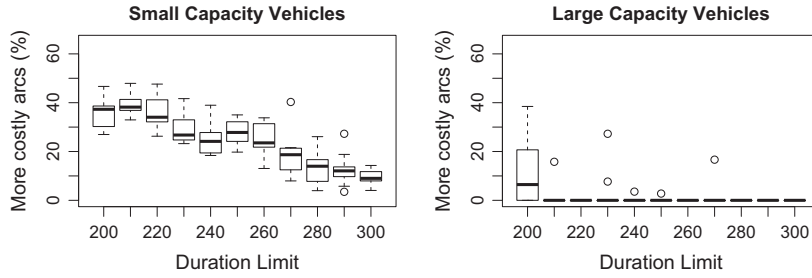


Fig. 9. The percentage of more costly (and shorter duration) arcs traversed by vehicle type.

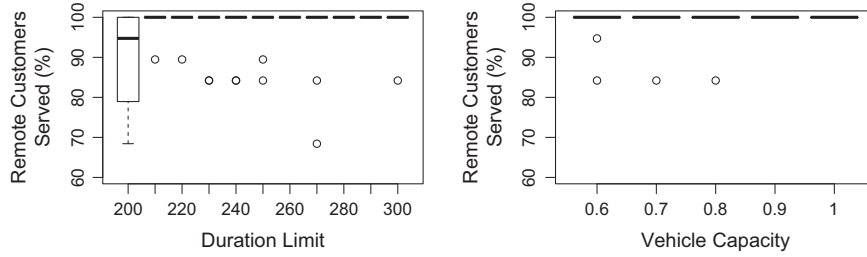


Fig. 10. The percentage of remote customers served by the small capacity vehicles.

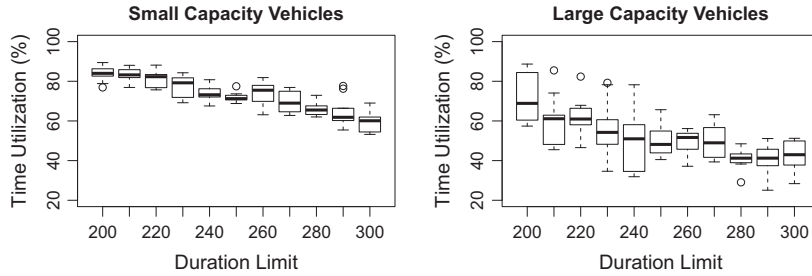


Fig. 11. Time utilization by vehicle type.

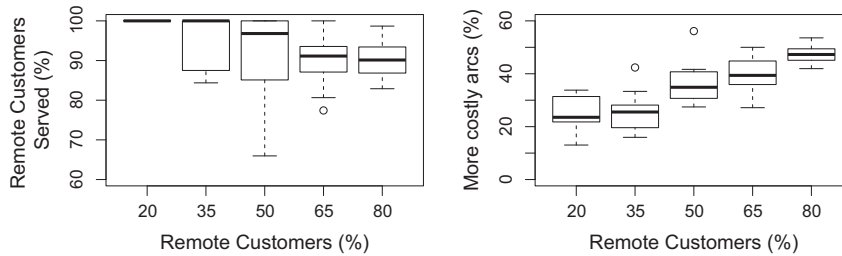


Fig. 12. Percentage of remote customers and more costly arcs in the routes of small capacity vehicles under different customer distributions.

In the experiments, we illustrate the percentage of remote customers served by the small capacity vehicles and the percentage of the more costly arcs in the routes of these vehicles. In addition, we show the time and capacity utilization with respect to different vehicle types. The capacity utilization is measured in a similar way to the time utilization, i.e., it is the ratio of total load delivered by the vehicle to its available capacity.

Figs. 12–14 summarize the findings. When more customers are located in the remote region, the small capacity vehicles tend to utilize a larger percentage of the more costly arcs, i.e., faster links, in order to serve a higher number of remote customers within the route duration limit. In fact, even when customers are uniformly distributed, the small capacity vehicles serve more than 80% of the customers in the remote region. Furthermore, the small capacity vehicles tend to have higher

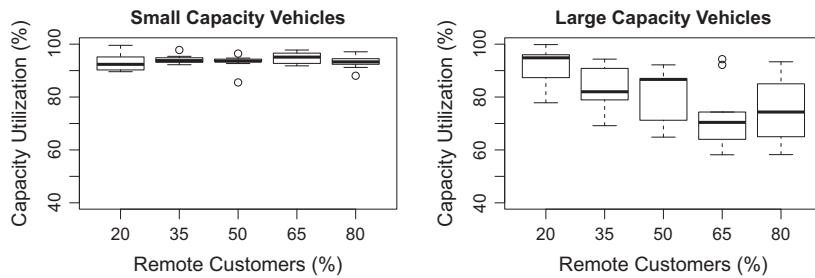


Fig. 13. Capacity utilization by vehicle type under different customer distributions.

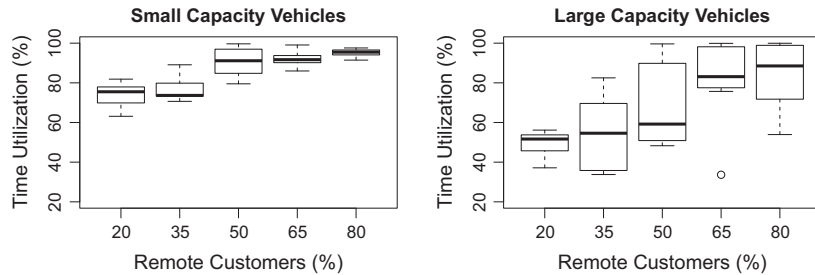


Fig. 14. Time utilization by vehicle type under different customer distributions.

time and capacity utilization than the large capacity vehicles. Given that the large vehicles are more costly, this suggests that the operating firm can achieve cost advantages and efficient use of resources by adapting their fleet size to have more of the smaller vehicles and by dispatching them to deliver the demand requests of the remotely-located customers.

6. Concluding remarks

Vehicle routing has been a central component in the operation of logistics and distribution systems. When determining the routes of a fleet of vehicles, decision makers may need to consider multiple criteria such as operational cost and customer service. For example, while the primary objective could be to minimize the total transportation cost, delivery time and frequency could be other important considerations. In the related literature, those criteria are often addressed by developing minimum-cost vehicle routes under time or distance constraints. These models can be used to understand the trade-off between the different criteria, depending on which decision makers can modify and customize vehicle trips. Another flexibility is related with the alternative paths of travel between the locations on the network. For example, vehicles can travel from an origin point to a destination point on a road network by following alternative routes of travel, e.g., faster travel at a higher cost or a longer distance. To consider the multiple attributes of these links in the solution space, it is necessary to expand the network representation of a vehicle routing problem by introducing parallel arcs between the vertices, which leads to a multigraph structure.

In this paper, we adopt the idea of alternative route consideration and study a time-constrained heterogeneous vehicle routing problem on a multigraph. We provide a mathematical formulation of the problem and develop a tabu search heuristic as the main solution approach. The presence of parallel arcs introduces computational challenges in exploring the neighborhood of a given solution, because, decisions regarding arc selection must be made in each iteration in addition to the assignment of a customer to a route. To this end, we propose a polynomial-time heuristic procedure for arc selection (FSAS) that helps us to search the neighborhood efficiently. By using instances with two parallel arcs and two attributes on each arc, we perform a numerical analysis and show that the proposed solution method is very effective in solving the problem. The tested instances illustrate the impact of alternative route consideration and also reveal some insights for the operation of the distribution system.

Our results suggest that considerable savings in transportation costs can be obtained by utilizing the alternative route structure, especially when the items must be delivered to customers in a restricted duration limit. This arises because the multigraph structure enables solutions that take advantage of the multiple attributes of the arcs, for example, an arc with a higher travel cost but a shorter travel time can be added to a vehicle route and this might reduce the need for additional vehicle dispatches. As practices such as just-in-time delivery and supply chain integration are putting pressure on firms to commit to quick deliveries, alternative routes can be utilized to gain benefits in transportation costs and flexibility. We also find that the parallel arc structure makes the highest impact on cost savings when the vehicles with smaller capacity are

dispatched to serve the customers in remote locations and the vehicles with larger capacity are utilized to serve customers nearby at a minimum cost. Overall, our model and numerical findings can be useful in developing effective dispatch policies for freight operators. For example, firms can generate higher savings in transportation costs and provide better customer service if they dispatch smaller capacity vehicles to serve more of the remote customers as these vehicles tend to take advantage of the alternative routes under restricted route duration limits.

Our work has some limitations, which suggest possible directions for future research. First, we have tested our proposed solution methodology on instances with two parallel arcs between vertex pairs and two attributes associated with these arcs. While the two attributes are likely to reasonably address most trade-offs such as in travel time and cost, a future study could focus on implementing the tabu search algorithm to handle the more general case of multiple arc attributes. We note that the tabu search and the FSAS procedure that we have developed are extendable to such a general case. In another line of research, the case in which not all parallel arcs are available for each vehicle type could be investigated, which necessitates modification on the design of our tabu search algorithm. Since we consider route duration constraints in the presence of service and traveling times, it would also be worthwhile to study the effect of time windows at customer locations. In such a setting, interesting insights about the gains due to parallel arcs can be generated by allowing multiple routes for each vehicle. For example, while the longer duration limits reduce the traveling costs because the less costly arcs are used more frequently (e.g., Fig. 7), the shorter duration limits can enable a vehicle to execute multiple routes and hence reduce the costs. In the presence of time windows it would be interesting to investigate which of the two mechanisms will dominate. Finally, although we have designed our numerical experiments by considering guidelines from practice, it would be interesting to test the developed policies by using real data. In such a study, additional aspects of distribution operations such as delays and changing customer requests could be considered to dynamically design the vehicle routes.

Acknowledgements

The authors thank Editor Prof. Jiuh-Biing Sheu and four anonymous referees for their constructive suggestions, which helped improve the content and presentation of this article. They also thank Prof. W.E.H. Dullaert for his valuable comments and feedback. The work of the third author is partially supported by GRF grant CUHK414313 from the Hong Kong Research Grants Council.

References

- Assad, A.A., 1988. Modeling and implementation issues in vehicle routing. In: Golden, B.L., Assad, A.A. (Eds.), *Vehicle Routing: Methods and Studies*. Elsevier Science, Amsterdam, pp. 7–45.
- Baldacci, R., Battarra, M., Vigo, D., 2008. Routing a heterogeneous fleet of vehicles. In: Golden, B.L., Raghavan, S., Wasil, E. (Eds.), *The Vehicle Routing Problem: Latest Advances and New Challenges*, Operations Research/Computer Science Interfaces, vol. 43. Springer, US, pp. 3–27.
- Baldacci, R., Bodin, L., Mingozzi, A., 2006. The multiple disposal facilities and multiple inventory locations rollon–rolloff vehicle routing problem. *Comput. Oper. Res.* 33 (9), 2667–2702.
- Baldacci, R., Mingozzi, A., 2009. A unified exact method for solving different classes of vehicle routing problems. *Math. Program.* 120, 347–380.
- Baldacci, R., Toth, P., Vigo, D., 2010. Exact algorithms for routing problems under vehicle capacity constraints. *Ann. Oper. Res.* 175, 213–245.
- Barbarosoglu, G., Ozgur, D., 1999. A tabu search algorithm for the vehicle routing problem. *Comput. Oper. Res.* 26 (3), 255–270.
- Barnes, G., Langworthy, P., 2003. The Per-mile Costs of Operating Automobiles and Trucks. Research Report MN/RC 2003-19. Minnesota Department of Transportation Office of Research Services.
- Brandão, J., 2011. A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem. *Comput. Oper. Res.* 38 (1), 140–151.
- Brandão, J., Mercer, A., 1997. A tabu search algorithm for the multi-trip vehicle routing and scheduling problem. *Eur. J. Oper. Res.* 100 (1), 180–191.
- Chen, F.Y., Goh, M., Lee, J., Ou, J., Sharafalia, M., Teo, C.P., Goh, P.G., Sivanathan, P., 2001. ST logistics: distributing consumer goods in China. *Int. Trans. Oper. Res.* 8 (2), 203–219.
- Choi, E., Tcha, D.W., 2007. A column generation approach to the heterogeneous fleet vehicle routing problem. *Comput. Oper. Res.* 34 (7), 2080–2095.
- Clarke, G., Wright, J.W., 1964. Scheduling of vehicles from a central depot to a number of delivery points. *Oper. Res.* 12, 568–581.
- 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.
- Cordeau, J.-F., Gendreau, M.F., Hertz, A., Laporte, G., Sormany, J.S., 2005. New heuristics for the vehicle routing problem. In: Langevin, A., Riepe, D. (Eds.), *Logistics Systems and Optimization*. Springer, US, pp. 279–297.
- Cordeau, J.-F., Laporte, G., Mercier, A., 2001. A unified tabu search heuristic for vehicle routing problems with time windows. *J. Oper. Res. Soc.* 52 (8), 928–936.
- Dai, J.J., Zhou, C., 2008. Beer distribution in China. Research Report. The Supply Chain Logistics Institute, The School of Industrial and Systems Engineering, Georgia Institute of Technology.
- Dantzig, G.B., Ramser, J.H., 1959. The truck dispatching problem. *Manage. Sci.* 6 (1), 80–91.
- Desrochers, M., Lenstra, J.K., Savelsbergh, M.W.P., Soumis, F., 1988. Vehicle routing with time windows: optimization and approximation. In: Golden, B.L., Assad, A.A. (Eds.), *Vehicle Routing: Methods and Studies*. Elsevier Science Publishers, B.V., North-Holland, pp. 65–84.
- Desrochers, M., Verhoog, T.W., 1991. A new heuristic for the fleet size and mix vehicle routing problem. *Comput. Oper. Res.* 18 (3), 263–274.
- Duhamel, C., Lacomme, P., Prodhon, C., 2012. A hybrid evolutionary local search with depth first search split procedure for the heterogeneous vehicle routing problems. *Eng. Appl. Artif. Intell.* 25, 345–358.
- Fisher, M.L., Huang, J., Tang, B.X., 1986. Scheduling bulk-pickup-delivery vehicles in Shanghai. *Interfaces* 16 (2), 18–23.
- Garaix, T., Artigues, C., Feillet, D., Josselin, D., 2010. Vehicle routing problems with alternative paths: an application to on-demand transportation. *Eur. J. Oper. Res.* 204 (1), 62–75.
- Gendreau, M., Hertz, A., Laporte, G., 1994. A tabu search heuristic for the vehicle routing problem. *Manage. Sci.* 40, 276–290.
- Gendreau, M., Laporte, G., Musaraganyi, C., Taillard, É.D., 1999. A tabu search heuristic for the heterogeneous fleet vehicle routing problem. *Comput. Oper. Res.* 26 (12), 1153–1173.
- Gendreau, M., Laporte, G., Potvin, J.-Y., 2002. Metaheuristics for the capacitated VRP. In: Toth, P., Vigo, D. (Eds.), *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, pp. 129–154.
- Gendreau, M., Potvin, J.-Y., Bräysy, O., Hasle, G., Løkketangen, A., 2007. Metaheuristics for the vehicle routing problem and its extensions: a categorized bibliography. In: Raghavan, S., Golden, B.L., Wasil, E.A. (Eds.), *The Vehicle Routing Problem: Latest Advances and Challenges*. Springer, pp. 143–170.

- Gheysens, F.G., Golden, B.L., Assad, A.A., 1984. A comparison of techniques for solving the fleet size and mix vehicle routing problem. *OR Spectrum* 6 (4), 207–216.
- Glover, F., 1986. Future paths for integer programming and links to artificial intelligence. *Comput. Oper. Res.* 13 (5), 533–549 (1).
- Glover, F., 1989. Tabu search Part I. *ORSA J. Comput.* 1 (3), 1–190.
- Golden, B.L., Raghavan, S.S., Wasil, E.A. (Eds.), 2008. *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer, New York.
- Golden, B.L., Assad, A.A., Levy, L., Gheysens, F.G., 1984. The fleet size and mix vehicle routing problem. *Comput. Oper. Res.* 11, 49–66.
- Hifi, M., Michrafy, M., Sbihi, A., 2004. Heuristic algorithms for the multiple-choice multidimensional knapsack problem. *J. Oper. Res. Soc.* 55, 1323–1332.
- Ho, S.C., Gendreau, M., 2006. Path relinking for the vehicle routing problem. *J. Heuristics* 12 (1–2), 55–72.
- Laporte, G., 1992. The vehicle routing problem: an overview of exact and approximate algorithms. *Eur. J. Oper. Res.* 59 (3), 345–358.
- Laporte, G., 2007. What you should know about the vehicle routing problem. *Naval Res. Logist.* 54 (8), 811–819.
- Laporte, G., 2009. Fifty years of vehicle routing. *Transp. Sci.* 43, 408–416.
- Laporte, G., Desrochers, M., Norbert, Y., 1984. Two exact algorithms for the distance-constrained vehicle routing problem. *Networks* 14 (1), 161–172.
- Laporte, G., Norbert, Y., Taillefer, S., 1988. Solving a family of multi-depot vehicle routing and location-routing problems. *Transp. Sci.* 22 (3), 161–172.
- Laporte, G., Norbert, Y., Desrochers, M., 1985. Optimal routing under capacity and distance restrictions. *Oper. Res.* 33 (5), 1050–1073.
- Leung, S.C.H., Zhang, Z., Zhang, D., Hua, X., Lim, M.K., 2013. A meta-heuristic algorithm for heterogeneous fleet vehicle routing problems with two-dimensional loading constraints. *Eur. J. Oper. Res.* 225 (2), 199–210.
- Li, C.-L., Simchi-Levi, D., Desrochers, M., 1992. On the distance constrained vehicle routing problem. *Oper. Res.* 40 (4), 790–799.
- Li, F., Golden, B.L., Wasil, E.A., 2007. A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem. *Comput. Oper. Res.* 34 (9), 2734–2742.
- Li, X., Tian, P., Aneja, Y.P., 2010. An adaptive memory programming metaheuristic for the heterogeneous fixed fleet vehicle routing problem. *Transp. Res. Part E: Logist. Transp. Rev.* 46 (6), 1111–1127.
- Liu, S., Huang, W., Ma, H., 2009. An effective genetic algorithm for the fleet size and mix vehicle routing problems. *Transp. Res. Part E: Logist. Transp. Rev.* 45 (3), 434–445.
- Ma, H., Cheang, B., Lim, A., Zhang, L., Zhu, Y., 2012. An investigation into the vehicle routing problem with time windows and link capacity constraints. *Omega* 40 (3), 336–347.
- Miller, C.E., Tucker, A.W., Zemlin, R.A., 1960. Integer programming formulation of traveling salesman problems. *J. ACM* 7 (4), 326–329.
- Nauss, R., 1978. The 0–1 knapsack problem with multiple choice constraints. *Eur. J. Oper. Res.* 2, 125–131.
- Ollie, B., Gendreau, M., 2005. Vehicle routing problem with time windows, Part I: route construction and local search algorithms. *Transp. Sci.* 39 (1), 104–118.
- Osman, I.H., 1993. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Ann. Oper. Res.* 41 (4), 421–451.
- Penna, P.H.V., Subramanian, A., Ochi, L.S., 2013. An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. *J. Heuristics* 19 (2), 201–232.
- Pessoa, A., de Aragão, M., Uchoa, E., 2009. A robust branch-and-cut-and-price algorithm for the heterogeneous fleet vehicle routing problem. *Networks* 54 (4), 167–177.
- Pisinger, D., Ropke, S., 2010. Large neighborhood search. In: Gendreau, M., Potvin, J.-Y. (Eds.), *Handbook of Metaheuristics*, International Series in Operations Research & Management Science, vol. 146. Springer, US, pp. 399–419.
- Potvin, J.-Y., 2009. State-of-the art review: evolutionary algorithms for vehicle routing. *INFORMS J. Comput.* 21, 518–548.
- Prins, C., 2009. Two memetic algorithms for heterogeneous fleet vehicle routing problems. *Eng. Appl. Artif. Intell.* 22 (6), 916–928.
- Renaud, J., Boctor, F.F., 2002. A sweep-based algorithm for the fleet size and mix vehicle routing problem. *Eur. J. Oper. Res.* 140 (3), 618–628.
- Renaud, J., Laporte, G., Boctor, F.F., 1996. A tabu search heuristic for the multi-depot vehicle routing problem. *Comput. Oper. Res.* 23 (3), 229–235.
- Rochat, Y., Taillard, É.D., 1995. Probabilistic diversification and intensification in local search for vehicle routing. *J. Heuristics* 1 (1), 147–167.
- Ropke, S., Pisinger, D., 2006. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transp. Sci.* 40 (4), 455–472.
- Salhi, S., Osman, I.H., 1996. Local search strategies for the vehicle fleet mix problem. In: Rayward-Smith, V.J., Osman, I.H., Reeves, C.R., Smith, G.D. (Eds.), *Modern Heuristic Search Methods*. Wiley, Chichester, pp. 131–154.
- Salhi, S., Rand, G.K., 1993. Incorporating vehicle routing into the vehicle fleet composition problem. *Eur. J. Oper. Res.* 66 (3), 313–330.
- Subramanian, A., Penna, P., Uchoa, E., Ochi, L., 2012. A hybrid algorithm for the heterogeneous fleet vehicle routing problem. *Eur. J. Oper. Res.* 221 (2), 285–295.
- Taillard, É.D., 1993. Parallel iterative search methods for vehicle routing problems. *Networks* 23 (8), 661–673.
- Taillard, É.D., 1999. A heuristic column generation method for the heterogeneous fleet VRP. *RAIRO – Oper. Res.* 33 (1), 1–14.
- Taillard, É.D., Laporte, G., Gendreau, M., 1996. Vehicle routing with multiple use of vehicles. *J. Oper. Res. Soc.* 47 (8), 1065–1070.
- Tarantilis, C.D., Kiranoudis, C.T., Vassiliadis, V.S., 2003. A list-based threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. *J. Oper. Res. Soc.* 54 (1), 65–71.
- Tarantilis, C.D., Kiranoudis, C.T., Vassiliadis, V.S., 2004. A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. *Eur. J. Oper. Res.* 152 (1), 148–158.
- Toth, P., Vigo, D., 1998. The granular tabu search and its application to the vehicle routing problem. *INFORMS J. Comput.* 15 (4), 333–346.
- Toth, P., Vigo, D. (Eds.), 2002. *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia.
- Vidal, T., Crainic, T.G., Gendreau, M., Prins, C., 2013. Heuristics for multi-attribute vehicle routing problems: a survey and synthesis. *Eur. J. Oper. Res.* 231, 1–21.
- Wassan, N.A., Osman, I.H., 2002. Tabu search variants for the mix fleet vehicle routing problem. *J. Oper. Res. Soc.* 53 (7), 738–782.
- Xu, J., Kelly, J.P., 1996. A network flow-based tabu search heuristic for the vehicle routing problem. *Transp. Sci.* 30 (4), 379–393.
- Xue, Z., Zhang, C., Lin, W.-H., Miao, L., Yang, P., 2014. A tabu search heuristic for the local container drayage problem under a new operation mode. *Transp. Res. Part E* 62, 136–150.
- Yaman, H., 2006. Formulations and valid inequalities for the heterogeneous vehicle routing problem. *Math. Program.* 106 (2), 365–390.
- Yu, M., Qi, X., 2014. A vehicle routing problem with multiple overlapped batches. *Transp. Res. Part E* 61, 40–55.