Transportation Research Part E 149 (2021) 102293

Contents lists available at ScienceDirect

Transportation Research Part E

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

A hybrid variable neighborhood search approach for the multi-depot green vehicle routing problem

Mir Ehsan Hesam Sadati *, Bülent Çatay

Faculty of Engineering and Natural Sciences, Sabanci University, Istanbul, Turkey
Smart Mobility and Logistics Lab, Sabanci University, Istanbul, Turkey

ARTICLE INFO

Keywords:
Green vehicle routing problem
Multi-depot
Variable neighborhood search
Tabu search
Refueling

ABSTRACT

The Multi-Depot Green Vehicle Routing Problem (MDGVRP) is an extension of the well-known Green Vehicle Routing Problem (GVRP) where a fleet of alternative fuel-powered vehicles (AFVs) are used to serve the customers. GVRP consists of determining AFV tours such that the total distance travelled is minimum. The AFVs depart from the depot, serve a set of customers, and complete their tours at the depot without exceeding their driving range and the maximum tour duration. AFVs may refuel en-route at public refueling stations. In MDGVRP, the AFVs are dispatched from different depot locations and may refuel during the day at any depot or refueling station. We formulate MDGVRP as a mixed integer linear programming model and develop a hybrid General Variable Neighborhood Search and Tabu Search approach by proposing new problem-specific neighborhood structures to solve the problem effectively. We assess the performance of our method using the GVRP dataset from the literature. Our results show that the proposed method can provide high quality solutions in short computation times. Then, we extend these instances to the multi-depot case and compare our solutions for small-size instances with the optimal solutions. We also report our results for large-size problems and investigate the trade-offs associated with operating multiple depots and adopting different refueling policies to provide further insights for both academicians and practitioners.

1. Introduction

Transportation activities are the largest source of greenhouse gas (GHG) emissions in the US and fourth largest globally (www.epa.gov). They were responsible for 27% of total (GHG) in the EU in 2017. GHG emissions had increased by 2.2% in 2017 compared to 2016, mainly because of higher emissions from road transport. Almost 72% of the total transport-related GHG emissions were caused by road transport, of which 9% were from light commercial vehicles and 19% from heavy-duty vehicles (www.eea.europa.eu). In comparison to its 1990 levels, GHG emissions should fall by around two thirds by 2050 in order to meet 60% emission reduction target of European Commission (EC Transport White Paper, 2011). So, the shift to cleaner vehicles in public and freight transport is a must in order to meet the targets and achieve the desired reductions in GHG emissions to avoid dangerous levels of global warming.

Nowadays, many logistic companies add their fleets with green vehicles that run with alternative fuel (such as ethanol, natural gas, electricity) instead of fossil fuel in order to lessen the adverse effects of their operations on the environment. Consequently, route planning of these alternative fuel vehicles (AFVs) has received increasing attention in the Vehicle Routing Problem (VRP) literature

* Corresponding author at: Faculty of Engineering and Natural Sciences, Sabanci University, Istanbul, Turkey.
E-mail address: msadati@sabanciuniv.edu (M.E.H. Sadati).

https://doi.org/10.1016/j.tre.2021.102293
Received 11 July 2020; Received in revised form 28 January 2021; Accepted 9 March 2021
Available online 31 March 2021

over the last decade. In the classical VRP, the fuel tank capacity of the vehicles is assumed unlimited since the vehicles can refuel easily and fast at any public gas station. However, green vehicles need alternative fuel stations (AFSs) for refueling, which are scarce (Koç and Karaoglan, 2016). The range anxiety of the AFVs and the scarcity of the AFSs make the resulting VRP more complex to model and solve.

The Green Vehicle Routing Problem (GVRP) was proposed by Erdoğan and Miller-Hooks (2012) as an extension of the Capacitated VRP (CVRP) where a fleet of AFVs was used to serve the customers instead of internal combustion engine vehicles (ICEVs). In this study, we address the Multi-Depot GVRP (MDGVRP), which is a relevant problem for many logistics companies that operate multiple regional depots for their last-mile delivery operations and employ AFVs to cut down their emissions and reduce fuel costs. In MDGVRP, the AFVs depart from the depot, serve a set of customers, and complete their tours at the depot without exceeding their driving range and the maximum tour duration. All customers must be served by exactly one AFV and AFVs may refuel at depots and public refueling stations when they run out of fuel. The objective is to minimize the total distance travelled. Note that the existence of multiple depots increases the difficulty of the problem significantly as each subset of customers assigned to a depot requires solving a separate GVRP. Since the utilization AFVs with limited autonomy makes the GVRP a challenging problem, its extension to MDGVRP brings even more complexity to the problem.

To solve this problem, we develop a hybrid algorithm that combines Variable Neighborhood Search (VNS) with Tabu Search (TS). The proposed algorithm applies TS with multiple, problem specific neighborhood structures as the local search mechanism within the general VNS (GVNS) scheme and is referred to as GVNS/TS. We perform an extensive experimental study to investigate the performance of the proposed method and provide insights for both researchers and practitioners. Various metaheuristic algorithms have been successfully employed for solving different VRP variants; however, VNS stands out with its simple algorithmic structure, ease of implementation, and good performance despite the use of very few parameters. Furthermore, allowing non-improving moves in a multiple neighborhood framework offers more potential for achieving high quality solutions compared to other simple local search methods such as Variable Neighborhood Descent (VND). These factors constitute the main motivation behind our methodology selection.

Our contributions may be summarized as follows: (i) we introduce MDGVRP and formulate its MILP model; (ii) we propose an efficient hybrid GVNS/TS method to solve it by implementing new problem specific neighborhood structures; (iii) we present an extensive computational study to evaluate the performance of the proposed method and provide managerial insights, (iv) we present new benchmark results to the literature. The remainder of this paper is organized as follows: Section 2 provides a review of related literature. Section 3 presents the mathematical formulation of the problem. Section 4 details the proposed GVNS/TS solution methodology. Section 5 presents the computational study and discusses the numerical results. Finally, conclusion remarks are provided in the last section.

2. Literature review

In the GVRP introduced by Erdoğan and Miller-Hooks (2012) the AFVs were allowed to refuel en-route in public refueling stations and the tank was assumed to be fully filled in constant time. The authors modelled the mixed integer linear programming (MILP) formulation of the problem where the objective is to minimize the total distance travelled and proposed two construction heuristics to solve it. Schneider et al. (2014) investigated the utilization of an electric vehicle (EV) fleet within the context of green vehicles and introducing the Electric VRP with Time Windows (EVRPTW). EVRPTW is similar to GVRP; however, the recharge duration was not constant and was a linear function of the total energy transferred. The battery was assumed to be fully recharged at the station. The authors proposed a Variable Neighborhood Search and Tabu Search approach, and solved the problems they generated based on Solomon (1987) data as well as the GVRP data of Erdoğan and Miller-Hooks (2012). Schneider et al. (2015) generalized GVRP by introducing the Vehicle Routing Problem with Intermediate Stops and proposed an Adaptive Variable Neighborhood Search (AVNS) to solve it. Bruglieri et al. (2016) proposed an efficient formulation of GVRP by implicitly considering the AFSs and reducing the graph by eliminating dominated AFSs that can be visited between each pair of customers.

Montoya et al. (2016) presented an efficient two-phase heuristic to tackle GVRP. In the first phase, they construct a pool of routes by employing a set of "route-first cluster-second" heuristics that optimally inserts the refueling stations in the routes. Then, in the second phase, they solve the set partitioning problem using the routes stored in the pool. Koç and Karaoglan (2016) developed a metaheuristic-based exact algorithm for solving GVRP. Their approach couples Simulated Annealing with a branch-and-bound method.

Leggieri and Haouari (2017) formulated a new mathematical model of GVRP and introduced a reduction mechanism. They also presented a practical solution approach that could solve medium-size instances optimally. Bruglieri et al. (2019) proposed a two-phase path-based solution method for solving GVRP. In the first phase, they generate all feasible paths, and then in the second phase, they combine them to generate the final routes by solving a MILP model. The authors also proposed dominance rules to eliminate the dominated paths and decrease the size of the problem.

Various variants of EVRPTW have been studied following the study of Schneider et al. (2014). In Keskin and Çatay (2016) and Bruglieri et al. (2015) the authors relaxed the full-recharge restriction, and proposed mathematical programming models and solution methods that allow partial recharge of the battery with any quantity. Desaulniers et al. (2016) developed a branch-price-and-cut algorithm to solve EVRPTW by adopting full and partial strategies allowing single and multiple recharges en-route.

In Goeke and Schneider (2015), the authors extended EVRPTW by considering a mixed fleet of internal combustion engine vehicles (ICEVs) and EVs. Hiermann et al. (2016) also considered a mixed fleet that consisted of different types of EVs. Montoya et al. (2017) proposed nonlinear charging functions whereas Felipe et al. (2014) allowed fast charging stations within the context of Electric VRP (EVRP) without customer time windows. Keskin and Çatay (2018) also investigated the impact of fast charging stations in EVRPTW and observed that they can significantly improve the route plans when customer time windows are narrow.

Other variants of EVRP addressed in the recent literature involve battery swapping (Yang and Sun, 2015; Hof et al., 2017), heterogeneous fleet (Macrina et al., 2019; Hiermann et al., 2019), location routing (Hof et al., 2017; Schiffer and Walther, 2017), nonlinear charging function (Froger et al., 2019), two-echelons (Jie et al., 2019), the effect of ambient temperature (Rastani et al., 2019), capacitated recharging stations (Froger et al., 2017; Keskin et al., 2019), stochastic waiting times at recharging stations (Keskin et al., 2021), and recharging during service (Cortés-Murcia et al., 2019).

In a recent study, Koyuncu and Yavuz (2019) presented two alternative formulations that involved various practical settings including refueling the vehicle at customer locations and stations, different refueling policies, mixed fleet, and investigated their performances. For a detailed review of the GVRP and EVRP literature we refer the reader to Erdelić and Carić (2019) and Asghari and Al-e (2020).

Although the research on GVRP has been rapidly growing in the transportation science and operations research community during the last decade, the logistics operations using multiple depots have attracted little attention. Multi-depot VRP (MDVRP) is an extensively studied problem in the VRP literature due to its practicality in many industries (Escobar et al., 2014b; Tu et al., 2014; Masmoudi et al., 2016; Zhen et al., 2020; Sadati et al., 2020a; Sadati et al., 2020b). A comprehensive review of MDVRP may be found in Montoya-Torres et al. (2015) and Ramos et al. (2020).

3. Problem formulation

Our model extends the GVRP formulation of Erdoğan and Miller-Hooks (2012) to the multiple-depot setting. For ease of understanding, we follow the same mathematical formulation and notation convention. Let $G = (V, A)$ denote a complete directed graph where V and A represent the set of vertices and set of arcs, respectively. V consists of three subsets: set of depots $D = \{v_1, v_2, \cdots, v_m\}$, set of customers $I = \{v_{m+1}, v_{m+2} \cdots, v_{m+n}\}$, and set of refueling stations $F = \{v_{m+n+1}, v_{m+n+2}, \cdots, v_{m+n+f}\}$. s_i is the service time (refuelling time) at customer (station) i. Similar to Erdoğan and Miller-Hooks (2012), we assume that the fuel tank of the vehicle is filled to full capacity and refueling time is constant. A nonnegative cost d_{ij} and travel time t_{ij} are associated with each arc (i, j), The fleet consists of identical vehicles with fuel consumption rate r and fuel tank capacity Q. The vehicles depart from and return to the same depot. The objective is to design a set of routes that visit each customer once such that the total travel distance is minimized. A maximum route duration T_{max} is imposed on the journey time of the vehicles. A refueling station may be visited multiple times by the same or different vehicles. So, for each potential visit we add a copy of the station and create $G' = (V', A')$, where $V' = V \cup F'$ and $F' = F \cup \{v_{m+n+f+1}, v_{m+n+f+2}, \cdots, v_{m+n+f+f'}\}$.

Fig. 1 depicts an illustrative example which involves 12 customers (C1-C12), two refuelling stations (S1 and S2), and two depots (D1 and D2). The depots can also be used for refuelling. We set $Q = 100$ and $r = 1$. The values next to the cylindrical shapes indicate the fuel level of the tank upon the arrival of the AFV at the corresponding node. AFV#1 departs from depot D1, serves customers C1 thru C3, and then returns to D1 for refueling. After refueling, it continues its route by visiting C4-C6 and returns to D1 at the end of its tour. AFV#2 departs from D2, serves C7 thru C9, refuels at S2 before visiting C10, and then returns to its depot D2. Finally, AFV#3 departs from D2, visits C11 and C12, and returns to D2 without needing any refuelling.

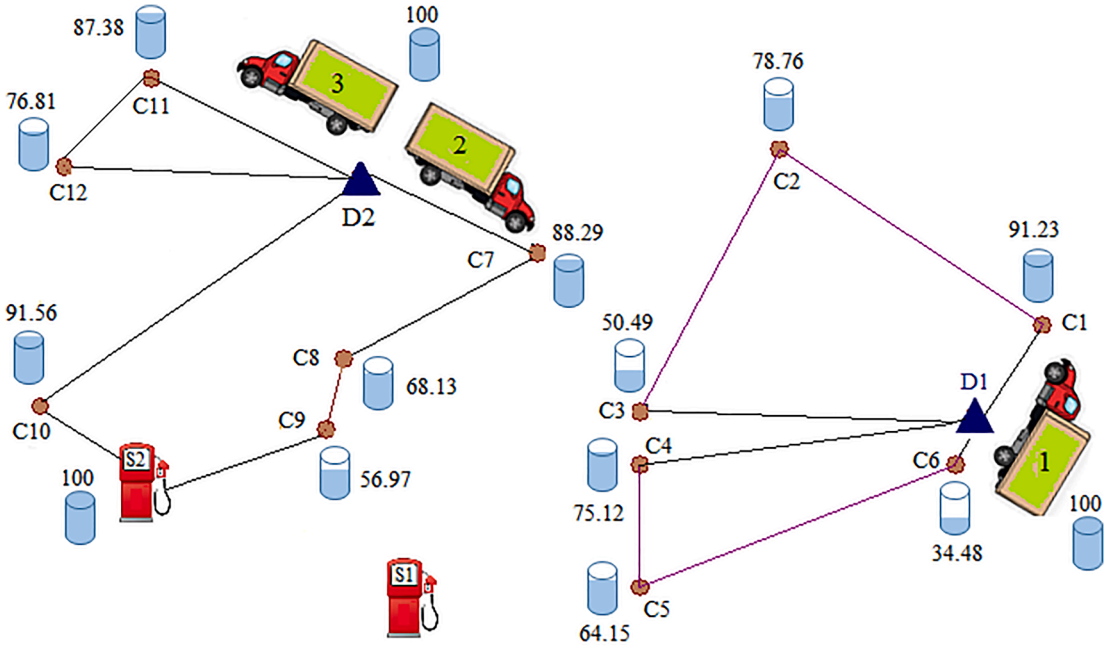

Fig. 1. An illustrative example of MDGVRP.

In what follows, we provide the MILP formulation of the problem. Table 1 summarizes the mathematical notation.

$$\min \sum_{i \in V'} \sum_{\substack{j \in V \\ i \neq j}} d_{ij} x_{ij} \tag{1}$$

subject to:

$$\sum_{\substack{j \in V' \\ i \neq j}} x_{ij} = 1, \qquad \forall i \in I \tag{2}$$

$$\sum_{\substack{j \in V' \\ i \neq j}} x_{ji} = 1, \qquad \forall i \in I \tag{3}$$

$$\sum_{j \in D} z_{ij} = 1, \qquad \forall i \in I \tag{4}$$

$$\sum_{\substack{j \in V' \\ i \neq j}} x_{ij} \leq 1, \qquad \forall i \in F' \tag{5}$$

$$\sum_{\substack{i \in V' \\ i \neq j}} x_{ij} - \sum_{\substack{i \in V' \\ i \neq j}} x_{ji} = 0, \qquad \forall j \in V' \tag{6}$$

$$x_{ij} \leq z_{ij}, \qquad \forall i \in R, \forall j \in D \tag{7}$$

$$x_{ji} \leq z_{ij}, \qquad \forall i \in R, \forall j \in D \tag{8}$$

$$x_{ik} + z_{ij} - z_{kj} \leq 1, \qquad \forall i \in R, \forall k \in R, i \neq k, \forall j \in D \tag{9}$$

$$x_{ik} + z_{kj} - z_{ij} \leq 1, \qquad \forall i \in R, \forall k \in R, i \neq k, \forall j \in D \tag{10}$$

$$\tau_j \geq \tau_i + s_i + t_{ij} - M\left(1 - x_{ij}\right), \qquad \forall i \in V', \forall j \in R, i \neq j \tag{11}$$

$$\tau_j \leq \tau_i + s_i + t_{ij} + M\left(1 - x_{ij}\right), \qquad \forall i \in V', \forall j \in R, i \neq j \tag{12}$$

Table 1
Mathematical notation.

Sets:	
I	Set of customers
D	Set of depots
F	Set of refueling stations
V	Set of depots, customers and refueling stations
F'	Set of all refueling stations including dummy stations
R	Set of customers and all refueling stations ($R = I \cup F'$)
V'	Set of all vertices ($V' = R \cup D$)
Parameters:	
d_{ij}	Distance from node i to node j
t_{ij}	Travel time from node i to node j
Q	Vehicle fuel tank capacity
r	Vehicle fuel consumption rate
s_i	Service time at node i (service time if $i \in I$, refueling time if $i \in F'$)
T_{max}	Maximum tour duration
M	A sufficiently large constant $\left(M = T_{max} + \underset{i,j \in R}{max}\{s_j + t_{ij}\}\right)$
Decision variables	
x_{ij}	1 if node j is visited following node i; 0 otherwise
z_{ij}	1 if node i is visited by a vehicle assigned to depot j; 0 otherwise
y_i	Vehicle fuel level upon its arrival at node i ($y_i = Q$ if $i \in F'$)
τ_i	Arrival time at node i $\left(\tau_j \geq s_j \text{ for } j \in D \text{ due to initial refueling at depot}\right)$.

$$\tau_i + s_i + t_{ij}x_{ij} \leq T_{max}\,, \qquad \forall i \in R, \forall j \in D \tag{13}$$

$$y_j \leq y_i - \left(r.\ d_{ij}\right)x_{ij} + Q\left(1 - x_{ij}\right), \qquad \forall i \in V^{'}, \forall j \in I, i \neq j \tag{14}$$

$$y_i \geq \sum_{\substack{i \in V^{'} \\ i \neq j}} \left(r.\ d_{ij}\right) x_{ij}, \qquad \forall i \in I \tag{15}$$

$$y_i = Q, \qquad \forall i \in F^{'} \tag{16}$$

$$x_{ij} \in \{0,1\}, \qquad \forall i.j \in V^{'} \tag{17}$$

$$z_{ij} \in \{0,1\}, \qquad \forall i \in R, \forall j \in D \tag{18}$$

$$y_i \geq 0, \qquad \forall i \in V^{'} \tag{19}$$

$$\tau_i \geq 0, \qquad \forall i \in V^{'} \tag{20}$$

The objective function (1) minimizes the total distance. Constraints (2) and (3) make sure that each customer is visited exactly once. Constraints (4) assigns each customer to exactly one depot. Constraints (5) restrict the visit to a station by at most once. Constraints (6) are the flow balance constraints. Constraints (7) and (8) guarantee that a vehicle travels from a node to a depot or from a depot to a node only if that node is assigned to that depot. Constraints (9) and (10) state that two nodes (customer or refueling station node) must be assigned to the same depot if they are visited consecutively by the same vehicle. Constraints (11)–(13) keep track of time and ensure that the maximum tour duration is not exceeded. They also prevent the formation of subtours. Constraints (14) keep track of tank fuel level at each customer while constraints (15) guarantee that the vehicle never runs out of fuel. Constraints (16) fill up the tank when the vehicle visits a refueling station. Finally, constraints (17)–(20) define the decision variables.

The Capacitated VRP (CVRP) is a special case of MDGVRP where a single depot is used instead of multiple depots and the fuel tank capacity (driving range) of the vehicles is unlimited ($Q = \infty$). Since CVRP is *NP-hard* (Toth and Vigo, 2002), MDGVRP is *NP-hard* as well.

4. Solution methodology

We present a hybrid GVNS/TS algorithm for solving MDGVRP. The basic component of the algorithm is General Variable Neighborhood Search (GVNS) developed by Mladenović and Hansen (1997). The neighborhood structures in GVNS/TS are changed systematically through the VNS approach and the local search is applied by implementing TS, which was originally developed by Glover (1986). VNS has a simple algorithmic structure and involves very few parameters; however, it is one of the most successful metaheuristics for solving hard combinatorial optimization problems. It has been applied to various routing problems such as the Electric VRP with Time Windws (Schneider et al., 2014), VRP with Simultaneous Pickup and Delivery with Time Limit (Polat et al., 2015), VRP with Two-Dimensional Loading Constraints (Wei et al., 2015) VRP with Divisible Deliveries and Pickups (Polat, 2017), Clustered VRP (Hintsch and Irnich, 2018), and the Dial-a-Ride Problem with Electric Vehicles (Masmoudi et al., 2018). The combination of VNS with TS was also used in the VRP context for solving the Capacitated Location Routing Problem (Escobar et al., 2014a), VRP with clustered backhauls and 3D loading constraints (Bortfeldt et al., 2015), VRP with Drones and En-route Operations (Schermer et al., 2019), and Long-Term Car Pooling Problem (Mlayah et al., 2020). A recent survey shows that TS and VNS are the two most frequently employed metaheuristic approaches for solving the VRP and its variants. We refer the reader to Elshaer and Awad (2020) for details.

4.1. General variable neighborhood search

In our proposed method, we first generate an initial solution S_0 by using Clarke and Wright (CW) savings algorithm (Clarke and Wright, 1964). An initial feasible solution for MDGVRP is created by first assigning customers to their nearest depot and then applying CW to each depot and the assigned customers. We implement the parallel version of CW where multiple vehicle routes are constructed simultaneously while respecting the driving range and tour duration limitations.

Next, we implement GVNS/TS by means of a set of neighborhood structures $N_k(k = 1, \ldots, k_{max})$ which comprises the shaking phase, and another set of neighborhood structures $M_l(l = 1, \ldots, l_{max})$ which comprises the local search phase. In shaking, solution $S^{'}$ is generated randomly in the first neighborhood N_1 of S_0, and then TS is performed by applying the first neighborhood l_1 to obtain a new solution $\overline{S}$. If $\overline{S}$ is feasible and its objective value is better than the incumbent solution S^*, S^* is replaced by $\overline{S}$ and TS is performed by continuing with neighborhood l_1. Otherwise, l is incremented by 1 and TS is performed for $\overline{S}$ by next neighborhood structure ($l = l + 1$). This procedure is continued until all TS neighborhood operators are explored ($l = l_{max}$) by restarting from neighborhood l_1 whenever an improved feasible solution is obtained. During the TS procedure, if a new incumbent solution is obtained, the index k is reset to 1 to start from the first shaking neighborhood operator. Otherwise, k is incremented by 1 ($k = k + 1$) and GVNS/TS restarts from the incumbent solution S^*. This procedure is repeated and terminates if the solution does not improve for a pre-specified number

of consecutive iterations (*MaxNonImp*) or a pre-specified limit on the number of iterations (*Iter*) has been reached. The pseudocode of GVNS/TS is presented in Algorithm 1.

Algorithm 1. (*The pseudocode of GVNS/TS*)

// Set of the neighborhood structures $N_k (k = 1, ..., k_{max})$ for shaking
// Set of the neighborhood structures $M_l (l = 1, ..., l_{max})$ for tabu search
1: Set $S_0 = S_{CW}$// Generate initial solution using Clarke and Wright savings algorithm
2: Set $S = S^* = S_0$
3: **repeat**
4: **for** $k = 1$ **to** k_{max} **do**
5: select a random solution S' from the k^{th} neighborhood $N_k(S)$ of S // Shaking
6: **for** $l = 1$ **to** l_{max} **do**
7: Find the best neighbor $\overline{S}$ of S' in $M_l(S')$ // Tabu Search
8: **if** $\overline{S}$ is feasible **and** $Z(\overline{S}) < Z(S^*)$
9: Update $S^* = \overline{S}$
10: Set $l = 1$
11: Set $k = 0$
12: **else**
13: Set $l = l + 1$
14: **end if**
15: Set $S' = \overline{S}$
16: **end for**
17: **if** $k = 0$
18: Set $k = 1$
19: **else**
20: Set $k = k + 1$
21: **end if**
22: Set $S = S^*$
23: **end for**
*24:***until** Stopping condition

4.2. Shaking

At each iteration of the shaking phase of GVNS/TS a random solution is generated using one of the neighborhood operators N_k. We use the following five neighborhoods: 1–0 *Move*, 1–1 *Exchange*, 2–2 *Exchange,* 1–2 *Exchange* and 1–1–1 *Exchange.*

1–0 *Move* and 1–1 *Exchange* are well-known neighborhood structures in the literature. The former removes a customer and inserts between two consecutive customer and the latter swaps two customer nodes. 2–2 *Exchange* swaps two pairs of consecutive customers whereas 1–2 *Exchange* swaps one customer with two consecutive customers. All four are applied as both *intra-route* and *inter-route* moves. 1–1–1 *Exchange* is a three-way inter-route operator that exchanges the positions of three customers on three different routes simultaneously. All of these neighborhood operators are explored in a cyclic sequential order starting from N_1 = 1–0 *Move* and following the order above.

4.3. Strategic oscillation for handling infeasible solutions

GVNS/TS is carried out using strategic oscillation of Glover (2000) which allows infeasible solutions that violate tour duration and vehicle driving range constraints. Strategic oscillation is based on the idea of accepting infeasible solution spaces in the hope of finding a better feasible solution in the following iterations and has been successfully employed in many studies (Cordeau et al., 1997; Toth and Vigo, 2003). It accepts such an infeasible solution by penalizing its objective function value as follows:

$$Z'(S) = Z(S) + \alpha Q(S) + \beta D(S) \tag{21}$$

$Z(S)$ refers to the value of objective function (1) associated with solution S and

$$Q(S) = \sum_{i \in V'} \sum_{j \in I} \left[y_j - y_i + \left(r.\ d_{ij} \right) x_{ij} - Q\left(1 - x_{ij}\right) \right]^+ \tag{22}$$

$$D(S) = \sum_{i \in R} \sum_{j \in D} \left[\tau_i + s_i + t_{ij} x_{ij} - T_{max} \right]^+. \tag{23}$$

where $Q(S)$ and $D(S)$ indicate the violation in driving range and maximum tour duration, respectively. α and β are positive penalty coefficients corresponding to infeasibilities, and $[\bullet]^+ = max\{0,\ \bullet\}$. If the current solution S is feasible, then we set $Z'(S) = Z(S)$. After each iteration, the penalty coefficients are updated to find feasible and infeasible solutions with approximately the same frequency. We set $\alpha = \alpha(1+\delta)$ if S is infeasible with respect to driving range constraint; and otherwise $\alpha = \frac{\alpha}{(1+\delta)}$. Likewise, if S is infeasible or feasible with respect to tour duration constraint, we set $\beta = \beta(1+\delta)$ or $\beta = \frac{\beta}{(1+\delta)}$, respectively, where δ is the parameter to update α and β.

4.4. Tabu search

TS with multiple neighborhood structures has been effectively employed in the literature (see e.g. Schneider et al., 2014; Soto et al., 2017; Qiu et al., 2018). So, at each iteration of GVNS/TS, we perform a local search by applying TS using four neighborhood operators: 2-*Opt*, 1-*AddStation*, 1-*DropStation* and 1-*SwapStation*. 2-*Opt* is a well-known operator whereas the others are problem specific. Considering each unused station, 1-*AddStation* examines all the arcs where it can be inserted and performs the best insertion. "Best insertion" here refers to the insertion of a station into an arc which decreases the cost of the current solution most or increases it least, if no cost improving insertion exists. In 1-*DropStation*, a station is removed from the route by connecting its predecessor node to its successor node. "Best removal" is performed such that the removal of the station from the solution decreases the cost of the current solution most or increases it least, if no cost improving removal exists. Finally, 1-*SwapStation* applies 1-*AddStation* and 1-*DropStation* simultaneously. All neighborhood operators are illustrated in Figs. 2-4.

TS is applied by exploring the first move l_1 and a new solution $\overline{S}$ is obtained from the current solution $S^{'}$. If $\overline{S}$ is a feasible solution with a better objective function value than the incumbent solution S^{*}, $\overline{S}$ will replaced by S^{*} and TS is performed by reinitializing the neighborhood index to one ($l = 1$). Otherwise, l is increased by 1 and TS starts from $\overline{S}$ with next neighborhood structure ($l = l + 1$). This procedure is repeated until all neighborhood operators are explored ($l = l_{max}$) in cycling order. The TS terminates afterη_{Tabu} iterations. These four neighborhood operators are explored in a cyclic sequential order starting from M_1 = 2-*Opt* and end with M_4 = 1-*SwapStation*.

In order to prevent cycling in exploring the solution space, tabu conditions are used associated with each neighborhood. The tabu condition of each move is defined as follows: A (the) node(s) whose position is (are) changed (swapped) by a move cannot be displace (re-swapped) by the same move.

The tabu condition is repealed at the end of tabu duartion λ or if the aspiration criterion is satisfied, i.e. if the incumbent solution is improved. Furthermore, we implement the diversification strategy of Gendreau et al. (2008) that aims at searching the unexplored regions of the solution space. This procedure is as follows: let $\overline{S}$ be the current and $\widehat{S}$ be the new solution obtained by the l^{th} neighborhood.

If the objective function value of the new solutions is greater than equal to the objective function value of current solution ($Z^{'}(\widehat{S}) \geq Z^{'}(\overline{S})$), then $\eta = \sqrt{nv}\left(\frac{\rho_{i_{M_l}}}{t}\right)\left(Z^{'}(\widehat{S}) - Z^{'}(\overline{S})\right)$ is added to $Z^{'}(\widehat{S})$ as a penalty, where v and n represent the number of routes and the number of customers visited in $\overline{S}$, respectively. $\rho_{i_{M_l}}$ indicates the number of times that customer/station i is moved by operator l until then and t is the total number of iterations carried out so far.

Since an exhaustive search of these neighborhoods require significant computational effort we implement granular neighborhood search of Toth and Vigo (2003). In granular search, we accept the arcs with shorter distances (cost) than the threshold value defined as $= \frac{Z}{(n+v)}$, where Z denotes the objective function value of the initial solution. The acceptable arcs are rebuilt after $2n$ iterations by updating the granularity threshold to $\varphi^{'} = \frac{Z^{'}}{(n^{'}+v^{'})}$, where $Z^{'}, v^{'}$ and $n^{'}$ are the objective function value of the best solution obtained by GVNS/TS, the number of routes, and number of customers, respectively.

5. Experimental study

In this section, we first tune the parameters of GVNS/TS by performing initial experiments using a subset of the GVRP instances generated by Erdoğan and Miller-Hooks (2012). Next, we evaluate the performance of GVNS/TS using the whole set of GVRP instances

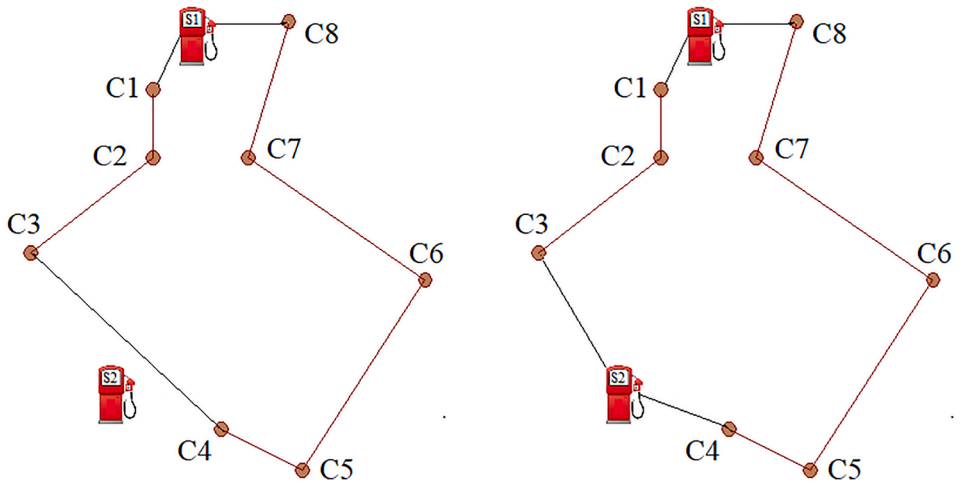

Fig. 2. 1-*AddStation* operator: (a) current solution, (b) solution after 1-*AddStation* move.

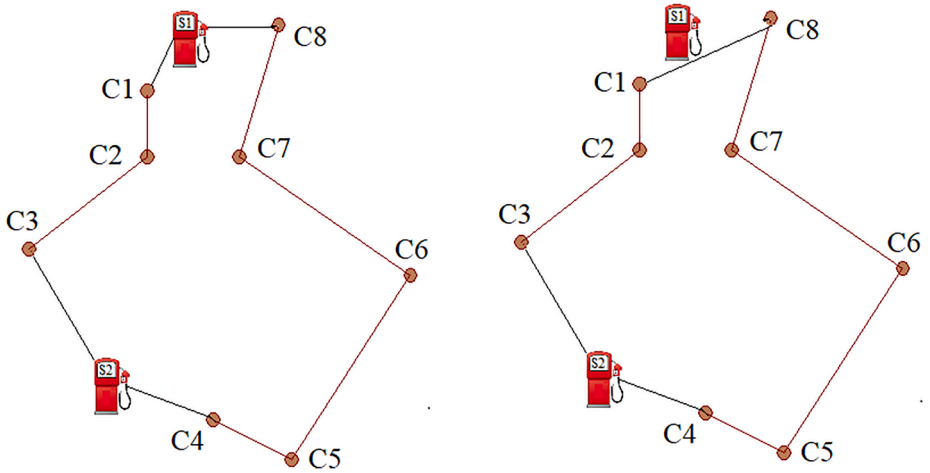

Fig. 3. 1-*DropStation* operator: (a) current solution, (b) solution after 1-*DropStation* move.

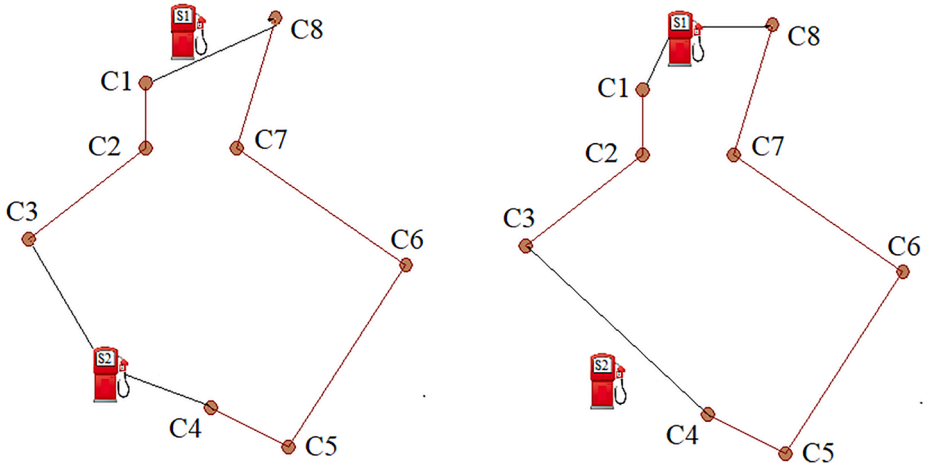

Fig. 4. 1-*SwapStation* operator: (a) current solution, (b) solution after 1-SwapStation move.

by comparing its results with those from the literature. Then, we modify the GVRP instances to create MDGVRP instances and solve them with GVNS/TS. For small-size data set, we use CPLEX to evaluate its performance. For large-size data set, we report our best solutions as benchmarks for future research. All experiments were carried out on a computer equipped with Intel Core i7-8700 3.2 GHz CPU and 32 GB RAM. The algorithm was coded in C# of Microsoft® Visual Studio 2019.

Table 2
Parameter tuning.

Parameter	Definition	Values (Δ%)					
λ	Tabu duration	1 (0.12)	2 (0.05)	5 (0.02)	**10 (0.00)**	20 (0.00)	40 (0.00)
α	Penalty for fuel capacity violation	0.2 (0.04)	0.3 (0.04)	**0.5 (0.00)**	1 (0.00)	2 (0.00)	–
β	Penalty for maximum tour duration violation	0.2 (0.03)	0.3 (0.02)	**0.5 (0.00)**	1 (0.00)	2 (0.00)	–
δ	Parameter to updateα, β	–	–	0.1 (0.09)	0.25 (0.01)	**0.5 (0.00)**	0.75 (0.00)
Iter	Total number of iterations	3000 (0.90)	5000 (0.42)	**10,000 (0.00)**	12,000 (0.00)	15,000 (0.00)	–
η_{Tabu}	Number of tabu search iterations	10 (0.14)	20 (0.08)	25 (0.08)	**50 (0.00)**	100 (0.00)	200 (0.00)
MaxNonImp	Maximum number of non-improving iterations	100 (0.26)	200 (0.10)	300 (0.05)	**500 (0.00)**	1000 (0.00)	2000 (0.00)

5.1. Parameter tuning

We selected 10 instances randomly from the GVRP data set and performed 10 runs for each by considering different values for each parameter. The selected instances are 20c3sU1-U2, 20c3sU9-U10, 20c3sC1-C2, 20c3sC9-C10, S1_10i6s, and S2_10i6s. For each parameter, we calculated the average percentage deviation (over 10 runs) from the best solution (Δ%) and set the value of the corresponding parameter to the value that produced the smallest Δ%. We repeated this procedure until parameters had been tuned. The parameters, their considered values, and the average percent deviation for each value are reported in Table 2. The values selected are indicated in bold. If two parameter values yielded the same deviation, we favored the smaller value.

5.2. Computational results for GVRP instances

We first tested the performance of GVNS/TS using the GVRP instances of Erdoğan and Miller-Hooks (2012) as well as Andelmin and Bartolini (2017) instances. In standard GVRP instances some customers may be infeasible and cannot be visited because of the given driving range (fuel tank capacity) of the vehicle and maximum tour duration constraints. Therefore, these infeasible customers are identified and discarded from the problem. So, we adopted the same approach.

Table 3
Results for small-size Erdoğan and Miller-Hooks data.

Instance	n	v	Optimal	GVNS/TS		SSG		SSH		MGMV	
				Best/Avg	t(s)[a]	t(s)[b]	Δ%	t(s)[b]	Δ%	t(s)[c]	Δ%
20c3sU1	20	6	1797.49	1797.49	1.26	41.40	0.00	9.60	0.00	4.80	0.00
20c3sU2	20	6	1574.77	1574.78	0.63	38.40	0.00	9.00	0.00	4.20	0.00
20c3sU3	20	6	1704.48	1704.48	0.63	38.40	0.00	7.80	0.00	4.20	0.00
20c3sU4	20	5	1482.00	1482.00	1.26	39.00	0.00	10.20	0.00	4.20	0.00
20c3sU5	20	6	1689.37	1689.37	1.26	40.20	0.00	10.80	0.00	4.20	0.00
20c3sU6	20	6	1618.65	1618.65	0.63	40.20	0.00	9.00	0.00	4.20	0.00
20c3sU7	20	6	1713.66	1713.66	1.26	38.40	0.00	11.40	0.00	4.20	0.00
20c3sU8	20	6	1706.50	1706.50	1.26	40.20	0.00	9.60	0.00	4.20	0.00
20c3sU9	20	6	1708.81	1708.82	0.63	39.60	0.00	11.40	0.00	4.20	0.00
20c3sU10	20	4	1181.31	1181.31	0.63	38.40	0.00	13.80	0.00	4.20	0.00
20c3sC1	20	4	1173.57	1173.57	0.63	37.20	0.00	22.80	0.00	4.20	0.00
20c3sC2	19	5	1539.97	1539.97	0.63	34.80	0.00	12.60	0.00	4.80	0.00
20c3sC3	12	3	880.20	880.20	0.63	15.00	0.00	9.00	0.00	2.40	0.00
20c3sC4	18	4	1059.35	1059.35	0.63	31.80	0.00	13.80	0.00	3.60	0.00
20c3sC5	19	7	2156.01	2156.01	0.63	36.00	0.00	8.40	0.00	6.00	0.00
20c3sC6	17	8	2758.17	2758.17	0.63	42.60	0.00	8.40	0.00	4.80	0.00
20c3sC7	6	4	1393.99	1393.99	0.02	10.80	0.00	2.40	0.00	3.60	0.00
20c3sC8	18	9	3139.72	3139.72	1.26	37.20	0.00	4.80	0.00	7.20	0.00
20c3sC9	19	6	1799.94	1799.94	0.63	36.00	0.00	9.60	0.00	6.00	0.00
20c3sC10	15	8	2583.42	2583.42	0.63	27.00	0.00	5.40	0.00	4.20	0.00
S1_2i6s	20	6	1578.12	1578.12	1.26	42.60	0.00	9.60	0.00	4.20	0.00
S1_4i6s	20	5	1397.27	1397.27	1.26	45.00	0.00	9.60	0.00	4.20	0.00
S1_6i6s	20	5	1560.49	1560.49	1.26	43.80	0.00	12.00	0.00	4.20	0.00
S1_8i6s	20	6	1692.32	1692.32	1.26	44.40	0.00	10.20	0.00	4.20	0.00
S1_10i6s	20	4	1173.48	1173.48	1.26	42.60	0.00	14.40	0.00	4.20	0.00
S2_2i6s	20	6	1633.10	1633.10	0.63	45.00	0.00	11.40	0.00	5.40	0.00
S2_4i6s	19	6	1505.07	1505.07	0.63	52.80	−1.82	8.40	0.00	5.40	0.00
S2_6i6s	20	7	2431.33	2431.33	1.89	46.80	0.00	7.80	0.00	4.20	0.00
S2_8i6s	16	7	2158.35	2158.35	1.26	34.20	0.00	5.40	0.00	3.60	0.00
S2_10i6s	16	5	1585.46	1585.46	1.26	36.60	−19.05	9.00	0.00	3.60	0.00
S1_4i2s	20	6	1582.20	1582.21	1.26	37.80	0.00	7.80	0.00	4.20	0.00
S1_4i4s	20	5	1460.09	1460.09	1.26	40.80	0.00	9.60	0.00	4.20	0.00
S1_4i6s5	20	5	1397.27	1397.27	1.26	45.00	0.00	9.60	0.00	4.20	0.00
S1_4i8s	20	5	1397.27	1397.27	1.26	49.20	0.00	10.20	0.00	4.20	0.00
S1_4i10s	20	5	1396.02	1396.02	1.89	51.00	0.00	13.80	0.00	4.20	0.00
S2_4i2s	18	4	1059.35	1059.35	0.63	30.60	0.00	13.80	0.00	3.60	0.00
S2_4i4s	19	5	1446.08	1446.08	0.63	36.00	0.00	12.60	0.00	5.40	0.00
S2_4i6s5	20	5	1434.14	1434.14	1.26	41.40	0.00	12.00	0.00	4.80	0.00
S2_4i8s	20	5	1434.14	1434.14	1.26	45.00	0.00	12.00	0.00	4.80	0.00
S2_4i10s	20	5	1434.13	1434.13	1.26	46.80	0.00	14.40	0.00	5.40	0.00
Average			1635.43	1635.43	1.26	39.00	−0.52	10.20	0.00	4.20	0.00

[a] Intel Core i7-8700 with 3.2 GHz (CPU Mark: 2696).
[b] Intel Core i5-750 with 2.67 GHz (CPU Mark: 1146).
[c] Intel Xeon E5410 with 2.33 GHz (CPU Mark: 997).

5.2.1. Numerical results on small-size GVRP instances

The small-size data set of Erdoğan and Miller-Hooks (2012) includes 40 instances involving 20 customers and the number of refueling stations vary between 2 and 10. Table 3 provides our results and compare them with those obtained by the state-of-the-art methods appeared in the literature. In this table, the first three columns "Instance", "n", and "v" provide the instance name, the number of customers, and fleet size, respectively. The column "Optimal" shows the optimal solutions reported by Bruglieri et al. (2019) whereas the three columns under "GVNS/TS" report our solutions where "Best", "Avg", and "$t(s)$" denote the best solution, average solution, and CPU time of the best solution in seconds, respectively. In the following columns, "SSG", "SSH", and "MGMV" refer to Hybrid Variable Neighborhood Search/Tabu Search of Schneider et al. (2014), Adaptive Variable Neighborhood Search of Schneider et al. (2015), and Multi-Space Sampling Heuristic of Montoya et al. (2016), respectively. The two columns "$t(s)$" and "Δ%" under each algorithm show the CPU time of the best solutions and percentage deviations of our solutions from those of the best solutions of the corresponding algorithms, respectively. All results are best of 10 runs and all CPU times are in seconds.

The results in Table 3 show that our proposed GVNS/TS was able to find the optimal solution in all of 40 instances. Moreover, the average and best solutions are same, i.e., GVNS/TS converged to the optimal solution in all 10 runs, which is a clear indication of the robustness of the algorithm. The comparison with SSG, SSH, MGMV reveals that our method outperforms SSG and performs as good as SSH and MGMV. Although the average results are not reported in the paper, we note that our overall average for 40 instances slightly outperforms SSH and MGMV by 0.15% and 0.01%, respectively (SSG did not provide their average results). On the other hand, our CPU times are significantly lower. While average run time is 1.26 s, the run times of SSG, SSH, and MGMV are 39.00, 10.20, and 4.20 s, respectively.

For a fair comparison, we specify the CPU (Single Thread Performance) used by each method and report its CPU Mark provided by Passmark below the table (www.cpubenchmark.net). The speed of the CPU is proportional to the reported value. According to these values, we see that our CPU is 2.52 times (15134/6007) faster than that of MGMV whereas our average run time is 3.34 times faster (4.20/1.26). Similarly, while our CPU is 4.07 times faster than that of SSG and SSH, our average run time is 30.95 and 8.10 times faster than that of SSG and SSH, respectively. So, we conclude that our proposed method outperforms all benchmark methods in terms of computational effort.

5.2.2. Numerical results on large-size GVRP instances

The large-size instances proposed by Erdoğan and Miller-Hooks (2012) involve 111 to 500 customers and 21 to 28 refueling stations. The results are given in Table 4. Similar to Table 3, the first three columns provide the instance name, the number of customers, and fleet size, respectively. The columns "BKS" and "Ref." indicate the best-known solution reported in the literature and the corresponding reference, respectively. The columns under "GVNS/TS" report our best solutions ("Best"), average solutions ("Avg"), the CPU times of the best solutions in seconds ("$t(s)$"), and percentage deviation of best ("Best Δ%"), and average ("Avg Δ%") solutions from BKSs, respectively.

We observe that the average deviation of the solutions obtained by GVNS/TS from BKS is only 0.13% in the best of 10 runs and 0.57% on the average of 10 runs. Our average CPU time is 794 s. So, we conclude that the proposed algorithm converges to very good solutions in reasonable computation times.

Next, we test the performance of our GVNS/TS using the dataset created by Andelmin and Bartolini (2017) by removing some customers from the large-size instances of Erdoğan and Miller-Hooks (2012). It consists of 40 instances involving 50 to 100 customers and categorized as AB1 and AB2 subsets. The customer and refueling stations are same in both subsets; however, the speed and fuel consumption rate of the vehicles in AB2 instances are 60 miles per hour (mph) and 0.2137 gallons per mile, respectively whereas they are set to 40 mph and 0.2 gallons per mile in AB1 and original data. Similar to original instances, some customers may be infeasible in the AB1 data set and cannot be visited because of the given driving range (fuel tank capacity) of the vehicle and maximum tour duration constraints. On the other hand, all customers are feasible in the AB2 data set.

In Table 5, we provide our results for all AB instances and compare them with the results reported by Andelmin and Bartolini (2017)

Table 4
Results for large-size Erdoğan and Miller-Hooks data.

					GVNS/TS				
Instance	n	v	BKS	Ref.	Best	Avg	t(s)	Best Δ%	Avg Δ%
111c_21s	109	17	4770.47	SSH	4772.43	4801.42	224	0.04	0.65
111c_22s	109	17	4774.65	MGMV	4778.29	4802.69	193	0.08	0.59
111c_24s	109	17	4767.14	SSH	4768.20	4785.26	168	0.02	0.38
111c_26s	109	17	4767.14	SSH	4768.20	4781.11	145	0.02	0.29
111c_28s	109	17	4765.52	SSH	4767.03	4779.81	194	0.03	0.30
200c_21s	192	31	8839.62	MGMV	8848.71	8902.73	395	0.10	0.71
250c_21s	237	37	10482.52	MGMV	10496.11	10548.74	476	0.13	0.63
300c_21s	283	44	12367.60	MGMV	12386.39	12452.28	977	0.15	0.68
350c_21s	329	50	14073.34	MGMV	14100.29	14180.61	1303	0.19	0.76
400c_21s	378	59	16660.20	MGMV	16695.89	16751.32	1543	0.21	0.55
450c_21s	424	65	18241.48	MGMV	18289.31	18332.08	1751	0.26	0.50
500c_21s	471	73	20496.50	MGMV	20562.18	20665.92	2153	0.32	0.83
Average			10417.18		10436.09	10482.00	794	0.13	0.57

Table 5
Results for Andelmin and Bartolini data.

Instance	n	v	AB		BMPP		GVNS/TS				
			Best	t(s)[a]	Best	t(s)[b]	Best	Avg	t(s)	Best Δ%	Avg Δ%
AB101	50	9	**2566.62**	1391	2584.88	211	**2566.62**	2571.05	13	0.00	0.17
AB102	50	10	**2876.26**	2266	2908.77	381	**2876.26**	2880.18	14	0.00	0.14
AB103	50	10	**2804.07**	1621	2869.37	379	**2804.07**	2807.35	14	0.00	0.12
AB104	47	9	**2634.17**	488	2660.33	185	2635.14	2637.70	16	0.04	0.13
AB105	73	14	**3939.96**	11082	3961.24	637	**3939.96**	3941.99	37	0.00	0.05
AB106	74	13	**3915.15**	3781	3953.20	610	3917.59	3920.40	27	0.06	0.13
AB107	75	13	**3732.97**	10905	3884.79	899	3737.18	3740.70	26	0.11	0.21
AB108	75	13	**3672.40**	4443	3760.92	609	3675.79	3678.30	20	0.09	0.16
AB109	75	13	**3722.17**	5679	3851.97	893	3723.29	3728.65	32	0.03	0.17
AB110	75	13	3612.95	9806	3753.00	796	3635.42	3640.81	38	0.62	0.77
AB111	71	14	**3996.96**	8484	4007.44	851	4003.28	4007.17	40	0.16	0.26
AB112	100	18	**5487.87**	11030	5844.05	2192	5495.82	5497.99	126	0.14	0.18
AB113	100	17	**4804.62**	12276	4980.98	1829	4810.16	4815.15	139	0.12	0.22
AB114	100	18	**5324.17**	11198	5505.54	2656	5330.29	5335.95	131	0.11	0.22
AB115	100	17	**5035.35**	13236	5338.85	2773	5040.68	5044.48	127	0.11	0.18
AB116	100	16	**4511.64**	11371	4600.60	2028	4520.19	4522.85	146	0.19	0.25
AB117	99	18	**5370.28**	12006	5439.55	2030	5380.49	5384.69	142	0.19	0.27
AB118	100	19	**5756.88**	13120	5857.64	2037	5760.19	5763.25	148	0.06	0.11
AB119	98	19	**5599.96**	11226	5682.33	2512	5601.29	5607.19	124	0.02	0.13
AB120	96	19	**5679.81**	11343	5846.51	1806	**5679.81**	5684.59	122	0.00	0.08
AB201	50	6	**1836.25**	1263	1882.75	423	**1836.25**	1839.52	21	0.00	0.18
AB202	50	6	**1966.82**	500	2008.90	443	**1966.82**	1969.66	16	0.00	0.14
AB203	50	6	**1921.59**	6149	1983.83	482	**1921.59**	1925.92	17	0.00	0.23
AB204	50	6	**2001.70**	1773	2048.12	367	**2001.70**	2006.03	26	0.00	0.22
AB205	75	9	**2793.01**	11056	2855.81	615	**2793.01**	2796.22	45	0.00	0.11
AB206	75	9	**2891.48**	12255	2976.43	594	**2891.48**	2894.74	35	0.00	0.11
AB207	75	8	**2717.34**	3145	2791.66	581	**2717.34**	2720.32	34	0.00	0.11
AB208	75	8	**2552.18**	3926	2619.92	411	2555.31	2559.09	20	0.12	0.27
AB209	75	8	**2517.69**	6814	2550.89	535	**2517.68**	2522.55	44	0.00	0.19
AB210	75	8	**2479.97**	9802	2538.98	573	**2479.89**	2482.05	33	0.00	0.08
AB211	75	9	2977.63	8358	3008.80	613	2979.19	2981.24	53	0.05	0.12
AB212	100	11	**3341.43**	11568	3409.74	1933	**3341.43**	3346.54	164	0.00	0.15
AB213	100	10	**3133.24**	11149	3207.99	1541	**3133.24**	3137.54	180	0.00	0.14
AB214	100	11	3384.28	12910	3457.66	1883	3384.28	3386.61	146	0.00	0.07
AB215	100	11	3480.52	12029	3537.11	2404	3481.98	3484.72	178	0.04	0.12
AB216	100	10	3221.78	13026	3301.22	1865	3222.16	3227.49	142	0.01	0.18
AB217	100	11	**3714.94**	12759	3797.22	1827	3715.19	3719.99	168	0.01	0.14
AB218	100	11	**3658.17**	14707	3727.27	1834	3661.29	3666.69	135	0.09	0.23
AB219	100	11	3790.71	12482	3862.63	1929	3795.09	3800.12	192	0.12	0.25
AB220	100	11	**3737.88**	11318	3807.88	1840	3741.26	3744.01	164	0.09	0.16
Average			3579.07	8594	3666.67	1225	3581.74	3585.54	82	0.06	0.18

and Bruglieri et al. (2019). In this table, the first three columns provide the instance name, the number of customers and fleet size, respectively. The following columns "AB" and "BMPP" refer to Andelmin and Bartolini (2017) and Bruglieri et al. (2019), respectively. Similar to the previous tables, the two columns under each algorithm show the best solution and CPU times of the corresponding algorithms in seconds; and the following five columns under "GVNS/TS" report our best solutions, average solutions, the CPU times of the best solutions in seconds, and percentage deviation of our best and average results from BKSs, respectively.

All results are best of 10 runs and all CPU times are in seconds. Note that Andelmin and Bartolini (2017) proposed an exact approach which provided the optimal solution in most of the instances. Bold values in AB and GVNS/TS "Best" columns indicate that the given result is optimal. The results show that GVNS/TS found the optimal solution in 16 instances. Moreover, the average deviation of the solutions obtained by GVNS/TS from those of AB is only 0.06% and 0.18% for the best and average of 10 runs, respectively. When we compare our results with those of BMPP we observe that GVNS/TS outperforms BMPP in all of instances and the average improvement is 2.22%. Furthermore, the average CPU time of GVNS/TS is 82 s compared to 8594 s of AB and 1225 s of BMPP. Although our CPU is faster these differences in run times are remarkable. Overall, these results show the superior performance of GVNS/TS in solving GVRP instances with regard to both computational time and solution quality. In what follows, we test its performance on small-size MDGVRP instances.

5.3. Computational results for MDGVRP instances

To generate MDGVRP instances we modified the GVRP instances proposed by Erdoğan and Miller-Hooks by converting some of refueling stations to depots. For small-size MDGVRP, we selected two depots, one is the original depot and the other is the first refueling station in the data. In large-size MDGVRP, we selected the first two refueling stations as depots in addition to the original

depot.

5.3.1. Results for small-size data

To test the performance of GVNS/TS, we solved the small-size MDGVRP instances on IBM ILOG CPLEX v.12.9.0 with a two-hour time limit. We used C# to call the CPLEX library using the same computer. The results are reported in Table 6. In this table, "m" represents the number of depots. Three columns under CPLEX indicate the upper bound ("UB"), percentage optimality gap ("Gap%"), and run time in seconds ("t(s)"), respectively. The columns "Best", "Best Δ%", "t(s)" under GVNS/TS indicate the best solution obtained after 10 runs of GVNS/TS, the percentage gap between our best solution and the CPLEX upper bound, and CPU time of the best solution in seconds, respectively. Similar to the GVRP instances, some customers in the MDGVRP problems cannot be served because of the limited driving range and maximum tour duration restrictions. So, these infeasible customers are discarded from the corresponding instance.

We observe that CPLEX could solve 14 instances out of 40 to optimality within the given time limit of two hours. The average optimality gap is 15.3% and the average run time 5311 s. GVNS/TS finds the optimal solution in 14 instances and match the CPLEX upper bound in 21 instances. In the reaming five instances GVNS/TS provided better solutions than the CPLEX upper bound (indicated in **bold**). Furthermore, the average computation time of GVNS/TS is only 4.27 s. Note that the deviation of the average of the average solutions of GVNS/TS from the CPLEX UB in 10 runs is only −0.04%. These results further validate the performance and robustness of the proposed GVNS/TS algorithm.

5.3.2. Results for large-size data

The results for 12 large-size MDGVRP instances are presented in Table 7. We report the best ("Best") and average ("Avg") solutions

Table 6
Results for small-size MDGVRP instances.

				CPLEX			GVNS/TS		
Instance	n	m	v	UB	Gap%	t(s)	Best	Best Δ%	t(s)
20c3sU1	20	2	6	1646.65	18.01	7200	1646.65	0.00	3.00
20c3sU2	20	2	5	1399.65	0.00	345	1399.65	0.00	3.31
20c3sU3	20	2	6	1640.55	0.00	2953	1640.55	0.00	3.48
20c3sU4	20	2	5	1338.35	0.00	5340	1338.35	0.00	3.27
20c3sU5	20	2	5	1395.60	11.10	7200	1395.60	0.00	2.46
20c3sU6	20	2	6	1402.06	0.00	1981	1402.06	0.00	2.97
20c3sU7	20	2	5	1518.18	0.00	2516	1518.18	0.00	3.45
20c3sU8	20	2	6	1493.79	0.00	5935	1493.79	0.00	3.89
20c3sU9	20	2	6	1663.46	25.69	7200	1663.46	0.00	3.95
20c3sU10	20	2	4	1115.89	0.00	51	1115.89	0.00	3.15
20c3sC1	20	2	4	1110.55	26.83	7200	1110.55	0.00	2.86
20c3sC2	19	2	5	1178.42	0.00	1564	1178.42	0.00	2.76
20c3sC3	12	2	3	880.20	0.00	16	880.20	0.00	1.90
20c3sC4	20	2	7	1640.45	21.12	7200	1640.45	0.00	3.56
20c3sC5	18	2	6	1760.20	69.66	7200	1760.20	0.00	3.49
20c3sC6	17	2	7	2270.48	9.31	7200	2213.44	**−2.51**	2.89
20c3sC7	6	2	3	911.01	0.00	1	911.01	0.00	0.80
20c3sC8	18	2	9	2852.19	0.00	897	2852.19	0.00	7.33
20c3sC9	19	2	5	1473.56	11.57	7200	1473.56	0.00	2.63
20c3sC10	18	2	9	2703.73	51.97	7200	2600.02	**−3.84**	6.19
S1_2i6s	20	2	6	1391.42	19.83	7200	1391.42	0.00	3.52
S1_4i6s	20	2	5	1194.63	10.09	7200	1194.63	0.00	3.86
S1_6i6s	20	2	5	1471.78	14.94	7200	1471.78	0.00	4.75
S1_8i6s	20	2	5	1393.45	8.70	7200	1393.45	0.00	3.14
S1_10i6s	20	2	4	1165.36	7.75	7200	1165.36	0.00	2.65
S2_2i6s	20	2	5	1509.96	36.59	7200	1509.96	0.00	5.52
S2_4i6s	19	2	4	1404.42	19.80	7200	1404.42	0.00	3.25
S2_466s	20	2	7	2341.91	0.00	389	2341.91	0.00	9.85
S2_8i6s	17	2	6	1554.22	15.46	7200	1509.22	**−2.90**	4.16
S2_10i6s	16	2	4	1625.28	48.11	7200	1625.28	0.00	5.78
S1_4i2s	20	2	6	1379.57	0.00	2038	1379.57	0.00	4.79
S1_4i4s	20	2	5	1263.85	0.00	1196	1263.85	0.00	2.99
S1_4i6s5	20	2	5	1194.63	10.04	7200	1194.63	0.00	4.09
S1_4i8s	20	2	5	1194.63	17.48	7200	1194.63	0.00	4.63
S1_4i10s	20	2	5	1194.63	19.43	7200	1194.63	0.00	5.71
S2_4i2s	20	2	7	1640.45	19.55	7200	1640.45	0.00	4.01
S2_4i4s	20	2	7	1640.35	21.33	7200	1640.35	0.00	4.31
S2_4i6s5	20	2	5	1315.61	28.09	7200	1315.61	0.00	7.12
S2_4i8s	20	2	5	1345.27	30.55	7200	1315.61	**−2.20**	8.75
S2_4i10s	20	2	5	1321.57	38.99	7200	1315.61	**−0.45**	10.72
Average				1498.45	15.30	5311	1492.41	**−0.30**	4.27

over 10 runs, CPU time of the best solution in seconds ("$t(s)$"), and the percentage deviation of the average solution from the best solution ("Δ%"). The CPU time on the average is 995 s and the deviation of the overall average of average solutions from best solutions is 0.21%. This small deviation shows the robustness of the proposed method in solving large-size problems.

5.3.3. Effect of using tabu search in local search

Allowing non-improving moves in a multiple neighborhood framework offers more potential for achieving high-quality solutions compared to those given by VND in the local search phase. To investigate the effect of TS on the performance of the algorithm, we repeated our tests by replacing TS with VND and referred to this algorithm as GVNS/VND. So, we re-solved all 12 large-size MDGVRP instances using GVNS/VND by removing the basic TS feature that allows non-improving moves and terminating the search at each neighborhood of 2-*Opt*, 1-*AddStation*, 1-*DropStation*, and 1-*SwapStation* when it has converged to the local optimum. These four neighborhoods are explored in a cyclic sequential order starting from $M_1 = 2$-*Opt* and end with $M_4 = 1$-*SwapStation*. While GVNS/VND stops exploring a neighborhood when the solution can no longer improve, GVNS/TS continues exploring that neighborhood by accepting non-improving moves if they are not tabu. The run time of GVNS/VND is expected to be shorther than that of GVNS/TS. Note that a perfect comparison is not possible because the CPU times differ from one run to another. To allow similar run times for a fair comparison, we increased the iteration number limit of GVNS/VND and set a time limit equal to the run time of the best solution obtained with GVNS/TS.

The results are provided in Table 8 where the columns "Best Δ%" and "Avg Δ%" show the percentage deviation of the GVNS/VND best and average objective function values over 10 runs from those of GVNS/TS. A positive Δ% value indicates that TS performs better than VND. Note that our setting slightly favored GVNS/VND with an overall average run time of 995 s compared to 968 s for GVNS/TS. Despite this drawback, the results show that GVNS/TS outperforms GVNS/VND in all of the instances and the contribution of TS to the solution quality is on average 1.74% and 2.22% for the best and average objective function values, respectively, as compared to the performance of VND. Furthermore, we observe that TS can enhance the performance of the algorithm by as much as 6.79% and 6.97% (instance 111c_22s) in the best and average solutions, respectively. From these results, we conclude that the algorithm greatly benefits from TS in achieving high-quality solutions.

5.4. Trade-off analysis

In this section, we consider five large-size MDGVRP instances (111c_21s, 200c_21s, 300c_21s, 400c_21s and 500c_21s) which involve the same stations but different number of customers, and investigate the trade-offs with respect to operating different numbers of depots and allowing refueling only at depots.

5.4.1. Influence of the number of depots

Originally, the number of depots is $m = 3$. Now, we consider $m = \{1, 2, 4, 5\}$ by keeping the original depot and assuming that the first $m-1$ refueling stations in the GVRP data are depots. Our aim is to investigate the trade-offs between operating more depots and several performance factors such as the number of customers served, fleet size and total distance traveled. Note that when $m = 1$, the problem reduces to standard GVRP.

The results are summarized in Fig. 5. Fig. 5(a) shows that increasing the number of depots improves the service level as more customers can be reached by the availability of additional depots. In other words, the driving range of AFVs is less restrictive in reaching the customers. Furthermore, in Fig. 5(b) we observe that a smaller fleet is needed when more depots are utilized.

For instance, in instance 500c_21s, increasing the number of depots from one to five decreases the fleet size from 73 to 47 vehicles. So, the cost of operating more depots can be compensated by using less AFVs. Finally, Fig. 5(c) shows how different number of depots affect the total distance (fuel cost) of the solution. The more the depots are operated the less is the total distance that the vehicles travel, despite the fact that more customers can be served with more depots. This is expected because increasing the number of depots allow customers located in that neighborhood to be served from their nearest depot. Overall, the results in Fig. 5 reveal the benefits of

Table 7
Results for large-size MDGVRP instances.

Instance	n	m	v	Best	Avg	t(s)	Δ%
111c_21s	109	3	14	3084.35	3084.35	278	0.00
111c_22s	109	3	14	3067.00	3069.18	293	0.07
111c_24s	109	3	14	3067.00	3067.00	269	0.00
111c_26s	109	3	14	3064.57	3066.12	269	0.05
111c_28s	109	3	14	3067.00	3067.00	282	0.00
200c_21s	194	3	23	5255.96	5260.89	653	0.09
250c_21s	240	3	27	6344.42	6380.92	801	0.58
300c_21s	289	3	35	8438.27	8470.21	1130	0.38
350c_21s	337	3	41	10170.07	10200.32	1340	0.30
400c_21s	386	3	46	11285.80	11312.87	1419	0.24
450c_21s	434	3	52	12610.96	12680.59	2304	0.55
500c_21s	483	3	59	14614.24	14647.20	2896	0.23
Average				7005.80	7025.55	995	0.21

Table 8
Comparison of TS and VND.

Instance	Best Δ%	Avg Δ%
111c_21s	2.40	4.01
111c_22s	6.79	6.97
111c_24s	2.82	3.93
111c_26s	1.22	1.69
111c_28s	0.94	1.54
200c_21s	1.69	2.57
250c_21s	0.46	0.58
300c_21s	0.63	0.66
350c_21s	0.60	0.77
400c_21s	1.12	1.56
450c_21s	0.87	0.97
500c_21s	1.29	1.37
Average	1.74	2.22

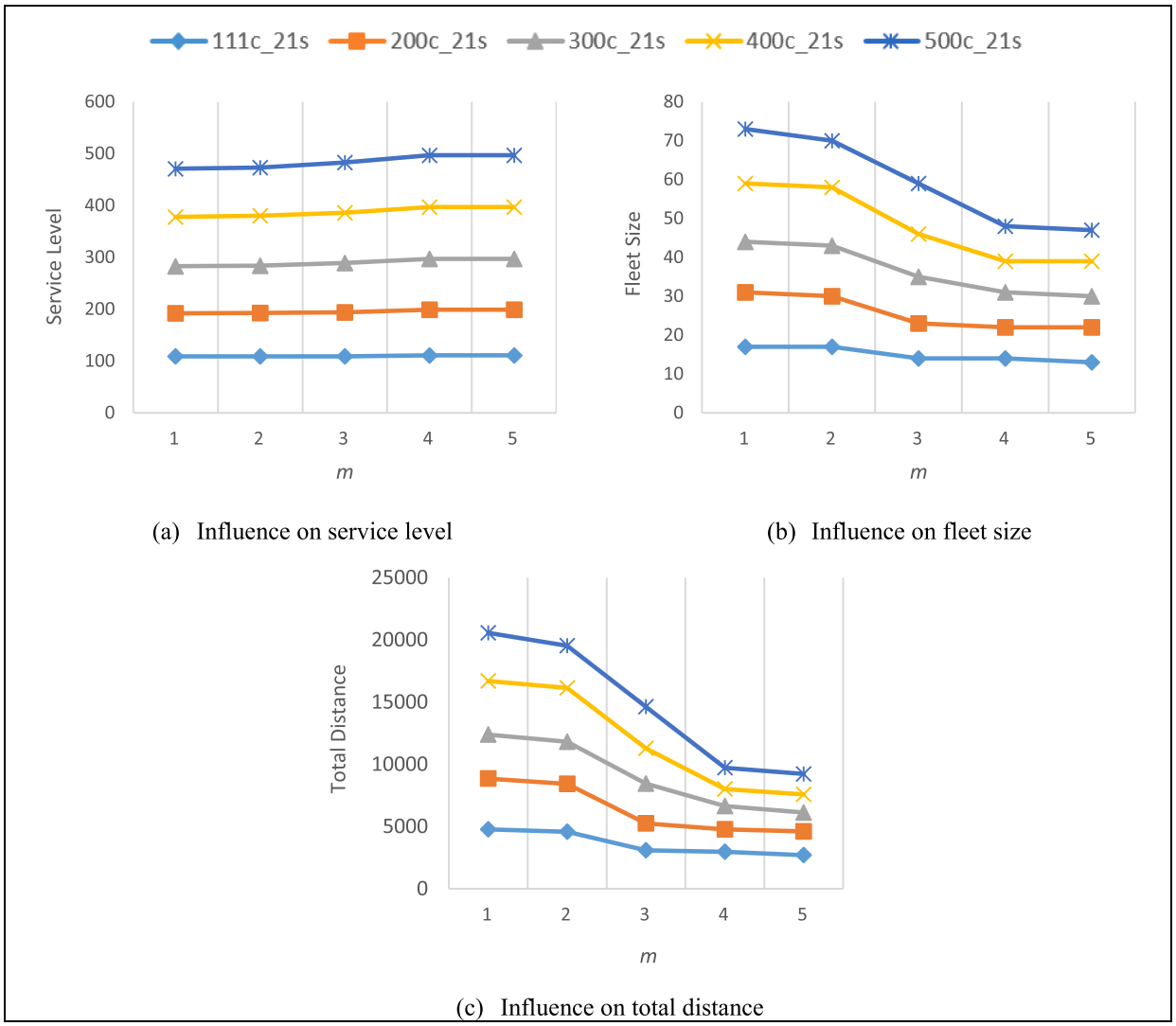

Fig. 5. Influence of different numbers of depots on performance factors.

operating multiple depots. However, a cost-benefit analysis is needed to assess the actual figures in a real business environment.

5.4.2. Influence of refueling only at depots

In this study, we allowed AFVs to refuel at depots and public stations. Even though numerous public AFSs may exist in the region, not all of them are truck friendly (Trey et al., 2016). In addition, many companies that employ AFVs prefer refueling them at their own facilities because of inefficient utilization of drivers' time and security concerns of their cargo (Morganti and Browne, 2018). So, refueling at depots may be the common situation in real-world logistics operations, considering the limited AFS infrastructures in most regions. In order to investigate the influence of refueling only at company-owned depots on the routing decisions we solved the same five MDGVRP instances discussed in the previous section by setting $m = 3$ and removing all refueling stations.

Fig. 6 summarizes the results according to the same performance factors, i.e. number of customers served, fleet size and total distance traveled. In this figure, the results are given in terms of the ratio of the performance factor values observed by allowing refueling at public AFSs vs. refueling at depots. For example, in the case of fleet size, the value provided represents $\left(\frac{v_{Stations}}{v_{depotsonly}}\right)$. In all instances of different sizes, we observe that all the ratios are greater than one. In other words, limiting the refueling at depots decreases the number of customers served, the fleet size and total distance. Even though the last two indicate improvements, a decrease in the number of customers suggests a lower service level. So, the reductions in fleet size and distance are achieved by visiting fewer customers. This is due the fact that the driving range of the AFVs does not permit service to some distant customers if the vehicles are not refueled at public stations available in their regions. Furthermore, we notice an increasing trend in the ratio values as the problem size gets larger. Since all five instances involve the same stations, we can conclude that the availability of public AFSs is a key business factor for logistics operators that serve hundreds of customers daily.

6. Concluding remarks

In this paper, we introduced a variant of the well-known Green Vehicle Routing Problem, namely Multi-Depot Green Vehicle Routing Problem where the customers are served from a set of depots using a fleet of AFVs. To solve the problem, we developed a hybrid GVNS/TS approach that combines the General Variable Neighborhood Search method (GVNS) with Tabu Search (TS). In our GVNS/TS, the neighborhood structures are systematically changing through the VNS approach and a TS-based local search procedure is implemented to exploit the search space. We validated the performance of the proposed method by solving GVRP instances and benchmarking our results to those of the state-of-the-art methods proposed in the literature. Then, we modified the GVRP data to generate MDGVRP instances and solved them using GVNS/TS. We compared the results of small-size problems against (near-)optimal solutions obtained with CPLEX. Our tests on both GVRP and MDGVRP data showed that GVNS/TS is robust and able to find high-quality solutions in reasonable computation times. We also provided managerial insights about the influence of operating different number of depots and allowing refueling only at depots on the routing decisions and customer service levels.

In this study, we assumed that the fuel tanks of the AFVs are filled to capacity in constant time. Future research on this topic may

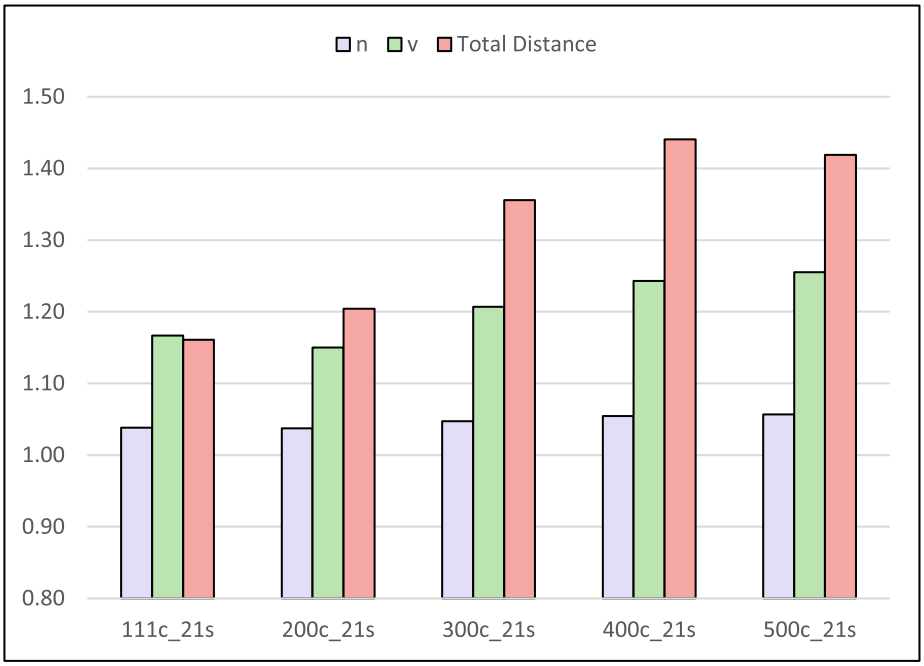

Fig. 6. Refueling at depots only vs. refueling at public stations.

address other realistic GVRP variants where different types of AFVs may be used with additional restrictions and limitations. Among those, Electric Vehicle Routing Problems (EVRPs) arise with challenging attributes such as long recharging durations at stations. The authors are planning to extend their method to solve EVRP with Time Windows as well as its multi-depot variants which would require new mechanisms to cope with service time window restriction and variable recharging time due to partial charging. Since it is not practical to convert the whole fleet to AFVs in short time, further research on this topic may focus on the utilization of a mixed fleet that may consist of ICEVs and different types of AFVs. Since vehicle types are associated with different total costs of ownership and fuel costs, the investigation of mixed fleets within this context may provide valuable managerial insights for logistics operators.

CRediT authorship contribution statement

Mir Ehsan Hesam Sadati: Conceptualization, Methodology, Software, Validation, Formal analysis, Writing - original draft, Visualization, Investigation, Data curation. **Bülent Çatay:** Conceptualization, Methodology, Validation, Formal analysis, Resources, Writing - original draft, Writing - review & editing, Data curation.

Acknowledgments

The authors wish to thank the three anonymous reviewers for providing valuable comments and suggestions throughout the review process of the paper.

References

Andelmin, J., Bartolini, E., 2017. An exact algorithm for the green vehicle routing problem. Transport. Sci. 51 (4), 1288–1303.
Asghari, M., Al-e, S.M.J.M., 2020. Green vehicle routing problem: a state-of-the-art review. Int. J. Prod. Econ.
Bortfeldt, A., Hahn, T., Männel, D., Mönch, L., 2015. Hybrid algorithms for the vehicle routing problem with clustered backhauls and 3D loading constraints. Eur. J. Oper. Res. 243 (1), 82–96.
Bruglieri, M., Pezzella, F., Pisacane, O., Suraci, S., 2015. A variable neighborhood search branching for the electric vehicle routing problem with time windows. Electron. Notes Discret. Math. 47, 221–228.
Bruglieri, M., Mancini, S., Pezzella, F., Pisacane, O., 2016. A new mathematical programming model for the green vehicle routing problem. Electron. Notes Discret. Math. 55, 89–92.
Bruglieri, M., Mancini, S., Pezzella, F., Pisacane, O., 2019. A path-based solution approach for the Green Vehicle Routing Problem. Comput. Oper. Res. 103, 109–122.
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., 1997. A tabu search heuristic for periodic and multi-depot vehicle routing problems. Networks 30 (2), 105–119.
Cortés-Murcia, D.L., Prodhon, C., Afsar, H.M., 2019. The electric vehicle routing problem with time windows, partial recharges and satellite customers. Transp. Res. Part E Logist. Transp. Rev. 130, 184–206.
Desaulniers, G., Errico, F., Irnich, S., Schneider, M., 2016. Exact algorithms for electric vehicle-routing problems with time windows. Oper. Res. 64 (6), 1388–1405.
Elshaer, R., Awad, H., 2020. A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants. Comput. Ind. Eng. 140, 106242.
Erdelić, T., Carić, T., 2019. A survey on the electric vehicle routing problem: variants and solution approaches. J. Adv. Transp. 2019.
Erdoğan, S., Miller-Hooks, E., 2012. A green vehicle routing problem. Transp. Res. Part E Logist. Transp. Rev. 48, 100–114.
Escobar, J.W., Linfati, R., Baldoquin, M.G., Toth, P., 2014a. A granular variable tabu neighborhood search for the capacitated location-routing problem. Transp. Res. Part B Methodol. 67, 344–356.
Escobar, J.W., Linfati, R., Toth, P., Baldoquin, M.G., 2014b. A hybrid granular tabu search algorithm for the multi-depot vehicle routing problem. J. Heuristics 20, 483–509.
European Commission White Paper on Transport. 2011. https://ec.europa.eu/transport/themes/european-strategies/white-paper-2011_en. (Accessed 28 January 2021).
Felipe, Á., Ortuño, M.T., Righini, G., Tirado, G., 2014. A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. Transp. Res. Part E Logist. Transp. Rev. 71, 111–128.
Froger, A., Mendoza, J.E., Jabali, O., Laporte, G., 2017. A matheuristics for the electric vehicle routing problem with capacitated charging stations. CIRRELT.
Froger, A., Mendoza, J.E., Jabali, O., Laporte, G., 2019. Improved formulations and algorithmic components for the electric vehicle routing problem with nonlinear charging functions. Comput. Oper. Res. 104, 256–294.
Gendreau, M., Iori, M., Laporte, G., Martello, S., 2008. A tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints. Networks 51, 4–18.
Glover, F., 2000. Multi-start and Strategic Oscillation Methods-principles to Exploit Adaptive Memory. Springer, Boston, MA, pp. 1–23.
Glover, F., 1986. Future paths for integer programming and links to artificial intelligence. Comput. Oper. Res. 13, 533–549.
Goeke, D., Schneider, M., 2015. Routing a mixed fleet of electric and conventional vehicles. Eur. J. Oper. Res. 245, 81–99.
Hiermann, G., Puchinger, J., Ropke, S., Hartl, R.F., 2016. The electric fleet size and mix vehicle routing problem with time windows and recharging stations. Eur. J. Oper. Res. 252, 995–1018.
Hiermann, G., Hartl, R.F., Puchinger, J., Vidal, T., 2019. Routing a mix of conventional, plug-in hybrid, and electric vehicles. Eur. J. Oper. Res. 272 (1), 235–248.
Hintsch, T., Irnich, S., 2018. Large multiple neighborhood search for the clustered vehicle-routing problem. Eur. J. Oper. Res. 270, 118–131.
Hof, J., Schneider, M., Goeke, D., 2017. Solving the battery swap station location-routing problem with capacitated electric vehicles using an AVNS algorithm for vehicle-routing problems with intermediate stops. Transp. Res. Part B Methodol. 97, 102–112.
Jie, W., Yang, J., Zhang, M., Huang, Y., 2019. The two-echelon capacitated electric vehicle routing problem with battery swapping stations: formulation and efficient methodology. Eur. J. Oper. Res. 272, 879–904.
Keskin, M., Çatay, B., 2016. Partial recharge strategies for the electric vehicle routing problem with time windows. Transp. Res. Part C Emerg. Technol. 65, 111–127.
Keskin, M., Çatay, B., 2018. A matheuristic method for the electric vehicle routing problem with time windows and fast chargers. Comput. Oper. Res. 100, 172–188.
Keskin, M., Laporte, G., Çatay, B., 2019. Electric vehicle routing problem with time-dependent waiting times at recharging stations. Comput. Oper. Res. 107, 77–94.
Keskin, M., Çatay, B., Laporte, G., 2021. A simulation-based heuristic for the electric vehicle routing problem with time windows and stochastic waiting times at recharging stations. Comput. Oper. Res. 125, 105060.
Koç, Ç., Karaoglan, I., 2016. The green vehicle routing problem: a heuristic based exact solution approach. Appl. Soft Comput. J. 39, 154–164.
Koyuncu, I., Yavuz, M., 2019. Duplicating nodes or arcs in green vehicle routing: a computational comparison of two formulations. Transp. Res. Part E Logist. Transp. Rev. 122, 605–623.
Leggieri, V., Haouari, M., 2017. A practical solution approach for the green vehicle routing problem. Transp. Res. Part E Logist. Transp. Rev. 104, 97–112.
Macrina, G., Di Puglia Pugliese, L., Guerriero, F., Laporte, G., 2019. The green mixed fleet vehicle routing problem with partial battery recharging and time windows. Comput. Oper. Res. 101, 183–199.

Masmoudi, M.A., Hosny, M., Braekers, K., Dammak, A., 2016. Three effective metaheuristics to solve the multi-depot multi-trip heterogeneous dial-a-ride problem. Transp. Res. Part E Logist. Transp. Rev. 96, 60–80.
Masmoudi, M.A., Hosny, M., Demir, E., Genikomsakis, K.N., Cheikhrouhou, N., 2018. The dial-a-ride problem with electric vehicles and battery swapping stations. Transp. Res. Part E Logist. Transp. Rev. 118, 392–420.
Mladenović, N., Hansen, P., 1997. Variable neighborhood search. Comput. Oper. Res. 24, 1097–1100.
Mlayah, I., Boudali, I., Tagina, M., 2020. A hybrid variable neighborhood tabu search for the long-term car pooling problem. In: Advances in Intelligent Systems and Computing. Springer Verlag, pp. 481–490.
Montoya, A., Guéret, C., Mendoza, J.E., Villegas, J.G., 2017. The electric vehicle routing problem with nonlinear charging function. Transp. Res. Part B Methodol. 103, 87–110.
Montoya, A., Guéret, C., Mendoza, J.E., Villegas, J.G., 2016. A multi-space sampling heuristic for the green vehicle routing problem. Transp. Res. Part C Emerg. Technol. 70, 113–128.
Montoya-Torres, J.R., Franco, J.L., Isaza, S.N., Jiménez, H.F., Herazo-Padilla, N., 2015. A literature review on the vehicle routing problem with multiple depots. Comput. Ind. Eng. 79, 115–129.
Morganti, E., Browne, M., 2018. Technical and operational obstacles to the adoption of electric vans in France and the UK: an operator perspective. Transp. Policy 63, 90–97.
Polat, O., 2017. A parallel variable neighborhood search for the vehicle routing problem with divisible deliveries and pickups. Comput. Oper. Res. 85, 71–86.
Polat, O., Kalayci, C.B., Kulak, O., Günther, H.O., 2015. A perturbation based variable neighborhood search heuristic for solving the Vehicle Routing Problem with Simultaneous Pickup and Delivery with Time Limit. Eur. J. Oper. Res. 242, 369–382.
Qiu, M., Fu, Z., Eglese, R., Tang, Q., 2018. A Tabu Search algorithm for the vehicle routing problem with discrete split deliveries and pickups. Comput. Oper. Res. 100, 102–116.
Rastani, S., Yüksel, T., Çatay, B., 2019. Effects of ambient temperature on the route planning of electric freight vehicles. Transp. Res. Part D Transp. Environ. 74, 124–141.
Ramos, T.R.P., Gomes, M.I., Póvoa, A.P.B., 2020. Multi-depot vehicle routing problem: a comparative study of alternative formulations. Int. J. Logist. Res. Appl. 23 (2), 103–120.
Sadati, M.E.H., Aksen, D., Aras, N., 2020a. The r-interdiction selective multi-depot vehicle routing problem. Int. Trans. Oper. Res. 27 (2), 835–866.
Sadati, M.E.H., Aksen, D., Aras, N., 2020b. A trilevel r-interdiction selective multi-depot vehicle routing problem with depot protection. Comput. Oper. Res. 104996.
Schermer, D., Moeini, M., Wendt, O., 2019. A hybrid VNS/Tabu search algorithm for solving the vehicle routing problem with drones and en route operations. Comput. Oper. Res. 109, 134–158.
Schiffer, M., Walther, G., 2017. The electric location routing problem with time windows and partial recharging. Eur. J. Oper. Res. 260 (3), 995–1013.
Schneider, M., Stenger, A., Goeke, D., 2014. The electric vehicle-routing problem with time windows and recharging stations. Transp. Sci. 48, 500–520.
Schneider, M., Stenger, A., Hof, J., 2015. An adaptive VNS algorithm for vehicle routing problems with intermediate stops. OR Spectr. 37, 353–387.
Solomon, M.M., 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. Oper. Res. 35 (2), 254–265.
Soto, M., Sevaux, M., Rossi, A., Reinholz, A., 2017. Multiple neighborhood search, tabu search and ejection chains for the multi-depot open vehicle routing problem. Comput. Ind. Eng. 107, 211–222.
Toth, P., Vigo, D. (Eds.), 2002. The Vehicle Routing Problem. Society for Industrial and Applied Mathematics.
Toth, P., Vigo, D., 2003. The granular tabu search and its application to the vehicle-routing problem. INFORMS J. Comput. 15, 333–346.
Trey, R.“, Baker, ”, Norboge, N., Moran, M., Wagner, J., Storey, B., 2016. Alternative Fuel Vehicle Forecasts Final Report. Texas A&M Transportation Institute.
Tu, W., Fang, Z., Li, Q., Shaw, S.L., Chen, B., 2014. A bi-level Voronoi diagram-based metaheuristic for a large-scale multi-depot vehicle routing problem. Transp. Res. Part E Logist. Transp. Rev. 61, 84–97.
Wei, L., Zhang, Z., Zhang, D., Lim, A., 2015. A variable neighborhood search for the capacitated vehicle routing problem with two-dimensional loading constraints. Eur. J. Oper. Res. 243, 798–814.
Yang, J., Sun, H., 2015. Battery swap station location-routing problem with capacitated electric vehicles. Comput. Oper. Res. 55, 217–232.
Zhen, L., Ma, C., Wang, K., Xiao, L., Zhang, W., 2020. Multi-depot multi-trip vehicle routing problem with time windows and release dates. Transp. Res. Part E Logist. Transp. Rev. 135, 101866.