TRANSPORTATION SCIENCE
Vol. 48, No. 4, November 2014, pp. 500–520
ISSN 0041-1655 (print) | ISSN 1526-5447 (online)

http://dx.doi.org/10.1287/trsc.2013.0490

The Electric Vehicle-Routing Problem with Time Windows and Recharging Stations

Michael Schneider
Logistics Planning and Information Systems, TU Darmstadt, 64289 Darmstadt, Germany,
schneider@bwl.tu-darmstadt.de

Andreas Stenger
Lufthansa Technik, 22335 Hamburg, Germany, andistenger@gmail.com

Dominik Goeke
Business Information Systems and Operations Research, University of Kaiserslautern,
67653 Kaiserslautern, Germany, dominik.goeke@wiwi.uni-kl.de

Driven by new laws and regulations concerning the emission of greenhouse gases, carriers are starting to use electric vehicles for last-mile deliveries. The limited battery capacities of these vehicles necessitate visits to recharging stations during delivery tours of industry-typical length, which have to be considered in the route planning to avoid inefficient vehicle routes with long detours. We introduce the electric vehicle-routing problem with time windows and recharging stations (E-VRPTW), which incorporates the possibility of recharging at any of the available stations using an appropriate recharging scheme. Furthermore, we consider limited vehicle freight capacities as well as customer time windows, which are the most important constraints in real-world logistics applications. As a solution method, we present a hybrid heuristic that combines a variable neighborhood search algorithm with a tabu search heuristic. Tests performed on newly designed instances for the E-VRPTW as well as on benchmark instances of related problems demonstrate the high performance of the heuristic proposed as well as the positive effect of the hybridization.

Keywords: electric vehicles; vehicle routing; hybrid metaheuristic; green logistics
History: Received: January 2012; revision received: January 2013; accepted: April 2013. Published online in *Articles in Advance* March 6, 2014.

1. Introduction

Present-day logistics strategies and operations may not be maintainable in the long run without serious negative effects on society and environment. Green logistics aims at improving the sustainability of production and distribution processes by taking into account environmental and social factors (Sbihi and Eglese 2010). The interest of companies in green logistics practices is driven by several factors such as, e.g., environmental regulations, rising energy and material costs, growing demands for green products and the potential of improving one's market position as a pioneer that starts to employ new green technologies and practices (see, e.g., Kleindorfer, Singhal, and Wassenhove 2005; Dekker, Bloemhof, and Mallidis 2012).

Transportation is one of the main addressees of green logistics activities because of its negative effects on the environment in the form of emissions, noise and congestion (Dekker, Bloemhof, and Mallidis 2012). For example, transportation activities are responsible for nearly 20% of greenhouse gas (GHG) emissions in the European Union (EU) in 2010 (EEA 2012), the lion's share caused by road transport (EC 2011a). Consequently, regulations on emissions are becoming increasingly popular. To reduce emissions of light commercial vehicles (<3.5t), EU regulation no. 510/2011 plans a penalty of 95 euro for each gram CO_2/km above 147 g CO_2/km of the manufacturers' average emissions starting in 2020 (EPEC 2011). The white book of the European Commission even envisages a mostly emission-free city logistics until 2030 (EC 2011b).

Distribution tasks of logistics companies are generally represented as vehicle routing problems (VRPs), which seek to minimize the total transportation costs of visiting a set of customers by means of multiple routes starting and ending at the depot. Over the years, many varieties and extensions of the VRP have been proposed to incorporate real-world constraints and conditions. Two of the most widely studied extensions are the capacitated VRP, where vehicles have a limited freight capacity (see, e.g., Toth and Vigo 2002), and the VRP with time windows (VRPTW), where customers have to be reached within a specified time interval (see, e.g., Gendreau and

Tarantilis 2010). To decrease the negative effects on the environment and meet emission standards, routing models and methods must integrate environmental considerations. One possibility is to incorporate emission costs into the objective functions of routing models, thus trading off environmental and economic goals (see, e.g., Figliozzi 2010; Bektaş and Laporte 2011). A different approach is the utilization of less polluting means of transport such as electric commercial vehicles (ECVs), whose specifics have to be included in adequate routing models. The contribution of our paper lies in this field.

ECVs are among the cleanest means of transport because they can be powered by sustainable and renewable energy sources (Dekker, Bloemhof, and Mallidis 2012; Davis and Figliozzi 2013), they have no local GHG emissions, and they produce only minimal noise. The latter two aspects are especially important in city areas with frequent traffic congestion (see, e.g., Figliozzi 2011). Moreover, ECVs allow meeting emission targets as they are defined to produce zero emissions (EU regulation no. 510/2011), and they may also serve as an instrument to achieve relative independence of global oil prices.

In earlier years, ECVs failed because of exorbitant battery prices and very short driving ranges. As ECVs have become one of the major research areas in the automotive sector, the magnitude of these problems diminishes. Although the replacement of conventional vehicles with ECVs is not profitable under most operation scenarios given the current cost conditions, the availability of increasingly long-lived batteries, rising fuel costs and lower ECV purchase costs are likely to change the picture (Davis and Figliozzi 2013). In the small-package shipping industry, several big companies, like DHL, UPS, and DPD have already started using ECVs for last-mile deliveries, particularly in urban areas (IPC 2011). Moreover, governments in all parts of the world promote the electrification trend and plan to provide the required infrastructure, e.g., a citywide electric-vehicle charging network goes into operation in London in 2013 (Transport for London 2013).

As mentioned earlier, a successful transition from conventional vehicles to ECVs necessitates efficient route-planning techniques, taking into account the specifics of ECVs. The maximum driving range of ECVs is estimated at 100–150 miles (Feng and Figliozzi 2013) but can be decreased significantly by cold temperatures and so-called range anxiety (Tredeau and Salameh 2009; Botsford and Szczepanek 2009). Thus, the available range is potentially not sufficient to perform the typical delivery tour of a logistics service provider in one run or to reach customers located far from the depot. Because reducing the number of deliveries performed by one vehicle is clearly not a profitable option, visits to recharging stations along the routes are required. These recharging visits have to be considered in the route planning to avoid inefficient vehicle routes with long detours, especially if the number of available recharging stations is scarce.

Moreover, routing models for ECVs have to include the most important practical constraints of logistics service provider that use ECVs for last-mile deliveries. First, vehicle capacity restrictions have to be considered for a significant share of delivery operations. Second, many companies, e.g., in the small-package shipping sector, face a high percentage of time-definite deliveries, which makes the integration of customer time windows into the routing model a necessity. The second aspect is especially interesting because recharging times for ECVs cannot be assumed to be fixed but depend on the current battery charge of the vehicle when arriving at the recharging station. Moreover, recharging operations take a significant amount of time, especially compared to the relatively short customer service times of, e.g., small-package shippers, and thus clearly affect the route planning.

In this paper, we introduce the electric vehicle-routing problem with time windows and recharging stations (E-VRPTW), which (1) incorporates the possibility of recharging at any of the available stations with recharging times depending on the charge level when arriving at the station, and (2) considers capacity constraints on vehicles as well as customer time windows. E-VRPTW aims at minimizing the number of employed vehicles and total traveled distance. Because E-VRPTW extends the well-known VRPTW, the high complexity of the problem renders exact solution methods inadequate for solving realistically-sized problem instances (Baldacci, Mingozzi, and Roberti 2012). To solve E-VRPTW, we develop a hybrid metaheuristic that combines a variable neighborhood search (VNS) heuristic with a tabu search (TS) method for the intensification phase of the VNS. In numerical studies, we prove the quality and efficiency of our VNS/TS on test instances of related problems, namely, the green VRP (Erdogan and Miller-Hooks 2012), the multidepot VRP with interdepot routes (Crevier, Cordeau, and Laporte 2007) and the standard VRPTW. Moreover, we design two sets of benchmark instances for E-VRPTW: a set of small-sized instances that we can solve exactly with the optimization software CPLEX to assess the performance of VNS/TS on E-VRPTW, and a set of more realistically-sized instances, on which we study the effectiveness of every component of our hybrid solution method.

The paper is organized as follows: In §2, a review of the related literature is presented. In §3, we introduce the notation in detail and provide a mixed-integer linear programming formulation of E-VRPTW. Section 4 describes the VNS/TS hybrid for solving E-VRPTW. Experimental results obtained on newly designed E-VRPTW instances as well as on benchmark sets of related problems are presented in §5. A short summary and conclusion of the paper are given in §6.

2. Literature Review

As described in the introduction, E-VRPTW strongly relates to the field of green logistics. Several overviews of the field highlighting different aspects of green logistics have been published in recent years (Kleindorfer, Singhal, and Wassenhove 2005; Srivastava 2007; McKinnon et al. 2010). The reviews of Sbihi and Eglese (2010) and Dekker, Bloemhof, and Mallidis (2012) address the operations research perspective of green logistics and the integration of environmental issues into combinatorial optimization problems.

Routing models considering alternative fuels, a limited driving range, and the possibility of recharging are quite scarce. Gonçalves et al. (2011) consider a VRP with pickup and delivery with a mixed fleet of ECVs and conventional vehicles. The total time to recharge an ECV is calculated from the total distance traveled and the possible travel range of one battery charge. However, they do not incorporate the actual location of recharging stations into their model. Conrad and Figliozzi (2011) present the recharging VRP, in which vehicles with limited range have the possibility of recharging en route at certain customer locations. The recharging time is assumed to be fixed. The impact of maximum driving range, recharging time, and time window existence is studied using a selection of the well-known Solomon instances. Moreover, bounds are formulated to predict average tour lengths. Erdogan and Miller-Hooks (2012) propose the green VRP (G-VRP), which considers a limited fuel capacity of the vehicles and the possibility to refuel at fuel stations along the route with a fixed refueling time. Neither capacity restrictions nor time window constraints are considered. The authors propose two heuristics to solve G-VRP. The first is a modified Clarke and Wright savings algorithm (MCWS) that creates routes by establishing feasibility through the insertion of fuel stations, merging feasible routes according to savings and removing redundant stations. The second heuristic is a density-based clustering algorithm (DBCA) designed as a cluster-first and route-second approach.

Our E-VRPTW and the works described previously integrate environmental issues in an indirect fashion by considering alternative fuels and their specifics, whereas the objective remains on cost optimization. Other routing problems incorporate green goals by optimizing objectives that are directly or indirectly related to the environment. Sbihi and Eglese (2010) highlight the indirect positive impact of the traditional VRP objective of minimizing traveled distance on emissions. Moreover, time-dependent VRPs are identified as indirectly decreasing emissions by avoiding congested routes. Maden, Eglese, and Black (2010) show in a case study concerned with scheduling a vehicle fleet in the United Kingdom, that the implicit consideration of congestion by means of time-dependent travel times leads to reduced CO_2 emissions of up to 7%. Figliozzi (2010) introduces the emissions VRP, which extends the VRPTW by incorporating the reduction of emissions into the objective function. The problem considers time-dependent travel times and involves driving speed and departure times as decision variables to design vehicle routes with minimal emissions. Jabali, Van Woensel, and de Kok (2012) propose the emissions-based time-dependent VRP with an objective function considering travel time, fuel and emission costs. The travel speed can be controlled in certain time periods during the day. The authors analyze the optimal speed in terms of minimizing emissions and present a TS method to solve the problem.

Bektaş and Laporte (2011) study the pollution-routing problem that extends the VRPTW and aims at minimizing a cost function involving emission, fuel and driver costs. Fuel consumption and emissions depend on vehicle speed, payload, grade and vehicle curb weight. Computational studies with CPLEX on several small-sized settings show a dominance of labor costs, but indicate potential to reduce the amount of emissions (although this does not necessarily minimize total costs). Demir, Bektaş, and Laporte (2012) extend the problem by considering the actual fuel consumption for low speeds, which is of high practical relevance for congested urban areas. To solve the problem, they propose an adaptive large neighborhood search extended by a speed optimization algorithm that computes optimal speeds with respect to the objective function.

Furthermore, several papers address case studies analyzing environmental aspects in the context of routing problems. Based on real-world travel speed data originating from the city of Portland, Figliozzi (2011) investigates the effect of different levels of congestion and speed limits on emissions. Ubeda, Arcelus, and Faulin (2011) present a case study on the introduction of green logistics practices in the transportation planning of a Spanish food company. The authors propose a new VRP that incorporates environmental aspects in both the objective function and in the routing logic. Faulin, Lera-López, and Juan (2011) propose extensions of traditional heuristics that consider estimates of safety and environmental costs from real-world scenarios in Spain to address the distribution problem of a Spanish food company.

The use of ECVs requires the integration of distance constraints depending on battery charge. Because of the widespread availability of petrol stations and the large cruising range of gasoline-powered vehicles, distance constraints have scarcely attracted interest as pure range (fuel) constraints (see, e.g., Laporte, Nobert, and Desrochers 1985; Li, Simchi-Levi, and Desrochers 1992; Kek, Cheu, and Meng 2008). Ichimori and Ishii (1981) propose an algorithm for solving a shortest-path problem in a network of refueling and standard nodes with fuel limitations. Ichimori, Ishii, and Nishida (1983) address the problem of determining the minimal range to reach all nodes without running out of fuel and present a polynomial algorithm to solve the problem. Some works on military issues propose concepts to extend the length of vehicle chains when fuel can be transferred between vehicles (Mehrez and Stern 1985; Melkman, Stern, and Mehrez 1986).

E-VRPTW is also related to other VRPs with intermediate stop requests because this is structurally similar to visits of recharging stations along the route. Crevier, Cordeau, and Laporte (2007) introduce the multidepot VRP with interdepot routes (MDVRPI), which considers intermediate depots at which vehicles can be replenished with goods during the course of a route. The authors develop a heuristic procedure that combines ideas from adaptive memory programming, described in Rochat and Taillard (1995), TS, and integer programming. Although the multidepot case is described, all proposed benchmark instances consider only one depot at which the vehicle fleet is stationed. Therefore, Tarantilis, Zachariadis, and Kiranoudis (2008) rename the problem to VRP with intermediate replenishment facilities (VRPIRF). They propose a hybrid-guided local search heuristic that follows a three-step procedure. First, an initial solution is constructed by means of cost-savings heuristics. Second, a VNS algorithm is applied using a TS in the local search phase. Third, the solution is further improved by means of a guided local search. Problems similar to MDVRPI also arise in the collection of waste, where collection vehicles have to visit a disposal facility as intermediate stop and can then continue their tour (for a recent review of the literature on waste collection, we refer to Beliën, De Boeck, and Van Ackere 2014).

Further works investigate the competitiveness of ECVs. Davis and Figliozzi (2013) examine the profitability of modern ECVs in comparison to conventional diesel trucks for different scenarios, based on a model for assessing energy consumption and total ownership cost. Long travel distances (still complying to the maximum driving range), slow speeds, frequent customer stops, low payloads, and long planning horizons are identified as dominant factors to render ECVs a viable alternative. A case study analyzing the competitiveness of ECVs in the U.S. market is presented in Feng and Figliozzi (2013). The authors find that the purchase of an ECV is only worthwhile if utilization is high and/or acquisition prices of ECVs decrease strongly. Kleindorfer et al. (2012) report practical insights from a project in which an analytical framework was designed as decision support for the fleet renewal and ECV adoption strategy of the French postal service La Poste.

The positioning of refueling stations has been discussed for vehicles using compressed natural gas (Boostani, Ghodsi, and Miab 2010) or electricity (He et al. 2013). Some models use a set-covering approach to determine number and location of refueling stations. Customers are usually aggregated to nodes to model the fuel demand of short-distance trips, whereas the flow between origin-destination pairs is used to model long-distance trips (Wang and Lin 2009; Wang and Wang 2010). Mak, Rong, and Shen (2013) study optimization models for locating battery-swapping stations and investigate the effect of battery standardization and technological progress on the optimal swapping infrastructure. Other work concentrates on finding the energy shortest path from a given origin to a destination, which can, e.g., be used in navigation systems. Given a battery capacity, the objective is to maximize the energy level at the destination while positive arcs represent energy consumption and negative arcs recuperation (Artmeier et al. 2010).

Finally, Wang and Shen (2007) propose a scheduling problem for electric buses where they assign timetabled trips that are known in advance to buses with the objective of minimizing total idle time. The travel range is limited by the vehicle's charge so that every vehicle has to be recharged after several trips. Beltran et al. (2009) present a model to assign a limited fleet of low- or zero-emission vehicles to lines of public transports.

3. The Electric Vehicle-Routing Problem with Time Windows and Recharging Stations (E-VRPTW)

Our model incorporates important characteristics of employing ECVs, namely a reduced operating range and the possibility to recharge at certain stations to increase this range. Following common VRP modeling techniques, we simplify several real-world characteristics and neglect the effect of payload on routing costs, assume a flat terrain, i.e., no grades are considered, and take travel speeds between vertices to be constant and given. Obviously, all of these factors influence the energy consumption of both ECVs and conventional vehicles. However, because the battery level (and thus the necessity to visit a recharging station and the resulting recharging time) depends on the energy consumption, the consideration of these characteristics seems even more important in the context of ECVs.

The energy consumption of real-world ECVs mainly depends on the power required to accelerate and to overcome rolling resistance, aerodynamic drag, and grade resistance (Davis and Figliozzi 2013). Because this power depends on the total weight of the vehicle, the payload of the vehicle or, more precisely, the distribution of the payload during the course of a route is important. Note that the assumption that the payload can be neglected in comparison to the curb weight of the vehicle approximately holds for some real-world applications, e.g., if the ECVs are used for small-package or letter delivery.

The aerodynamic drag depends quadratically on the speed of the vehicle, and therefore the speed at which the vehicle travels (or even the actual speed profile caused by congestion and the number of stops and signals) could be considered (Figliozzi 2011). Moreover, vehicle speed can be increased to fulfill time window requirements and reduced to decrease energy consumption, and could thus represent a decision variable. Speed profiles and grades are also important because of an ECV's ability to partially recover the kinetic energy lost by deceleration processes, called regenerative breaking. Furthermore, the power consumption does not directly translate to battery energy consumption because of the nonlinear efficiency of battery and electric drive train, which depends on the actual power demand (see, e.g., Guzzella and Sciarretta 2005; Zhang and Mi 2011). The integration of the described real-world characteristics into routing models for ECVs presents interesting opportunities for future study.

We formulate the E-VRPTW as a mixed-integer program that is used in the numerical studies to determine optimal solutions of small-sized instances by means of a commercial solver. Let V' be a set of vertices with $V' = V \cup F'$, where $V = \{1, \ldots, N\}$ denotes the set of customers and F' a set of dummy vertices generated to permit several visits to each vertex in the set F of recharging stations. Vertices 0 and $N+1$ denote the same depot, and every route starts at 0 and ends at $N+1$. To indicate that a set contains the respective instance of the depot, the set is subscripted with 0 and/or $N+1$, i.e., $F'_0 = F' \cup \{0\}$, $V'_0 = V' \cup \{0\}$, $V'_{N+1} = V' \cup \{N+1\}$ and $V'_{0,N+1} = V' \cup \{0\} \cup \{N+1\}$.

Thus, E-VRPTW can be defined on a complete directed graph $G = (V'_{0,N+1}, A)$, with the set of arcs $A = \{(i,j) \mid i, j \in V'_{0,N+1}, i \neq j\}$. With each arc, a distance d_{ij} and a travel time t_{ij} are associated. Each traveled arc consumes the amount $h \cdot d_{ij}$ of the remaining battery charge of the vehicle traveling the arc, where h denotes the constant charge consumption rate.

A set of homogeneous vehicles with a maximal capacity of C is positioned at the depot. Each vertex $i \in V'_{0,N+1}$ is assigned a positive demand q_i, which is 0 if $i \notin V$. Moreover, a time window $[e_i, l_i]$ in which service has to start is associated with each vertex $i \in V'_{0,N+1}$, and all vertices $i \in V_{0,N+1}$ have a service time s_i ($s_0, s_{N+1} = 0$). Service cannot begin before e_i, which might cause waiting time, and is not allowed to start after l_i, but might end later. At a recharging station, the difference between the present charge level and the battery capacity Q is recharged with a recharging rate of g, i.e., the recharging time incurred depends on the energy level of the vehicle when arriving at the respective station. For simplification reasons, we assume a linear recharge, although in real-world recharging processes the charging time increases for the last 10%–20% of the battery capacity (Marra et al. 2012).

Instead of a three-index formulation, we use decision variables associated with vertices to keep track of vehicle states, thus keeping the number of required variables low. Variable τ_i specifies the time of arrival, u_i the remaining cargo, and y_i the remaining charge level on arrival at vertex $i \in V'_{0,N+1}$. The decision variables x_{ij}, $i \in V'_0$, $j \in V'_{N+1}$, $i \neq j$ are binary and equal 1 if an arc is traveled and 0 otherwise. The variables and parameters of our model are summarized in Table 1. The objective function of E-VRPTW is hierarchical. As commonly done for VRPs with time window constraints (see, e.g., Bräysy and Gendreau 2005), our first objective is to minimize the number of vehicles, i.e., a solution with fewer vehicles is always superior. The second objective is to minimize the total traveled distance. Considering the high acquisition costs of ECVs, this objective seems even more reasonable than in the setting with conventional vehicles.

The mathematical model of E-VRPTW is formulated as a mixed-integer program as follows:

$$\min \sum_{i \in V'_0,\, j \in V'_{N+1},\, i \neq j} d_{ij} x_{ij} \tag{1}$$

$$\sum_{j \in V'_{N+1},\, i \neq j} x_{ij} = 1 \quad \forall\, i \in V, \tag{2}$$

$$\sum_{j \in V'_{N+1},\, i \neq j} x_{ij} \leq 1 \quad \forall\, i \in F', \tag{3}$$

$$\sum_{i \in V'_{N+1},\, i \neq j} x_{ji} - \sum_{i \in V'_0,\, i \neq j} x_{ij} = 0 \quad \forall\, j \in V', \tag{4}$$

$$\tau_i + (t_{ij} + s_i) x_{ij} - l_0(1 - x_{ij}) \leq \tau_j \qquad \forall\, i \in V_0,\ \forall\, j \in V'_{N+1},\ i \neq j, \tag{5}$$

$$\tau_i + t_{ij} x_{ij} + g(Q - y_i) - (l_0 + gQ)(1 - x_{ij}) \leq \tau_j \qquad \forall\, i \in F',\ \forall\, j \in V'_{N+1},\ i \neq j, \tag{6}$$

Table 1 Variable and Parameter Definitions of the E-VRPTW Model

$0, N+1$	Depot instances
F'	Set of visits to recharging stations, dummy vertices of set of recharging stations F
F'_0	Set of recharging visits including depot instance 0
V	Set of customers $V = \{1, \ldots, N\}$
V_0	Set of customers including depot instance 0
V'	Set of customer vertices including visits to recharging stations, $V' = V \cup F'$
V'_0	Set of customers and recharging visits including depot instance 0: $V'_0 = V' \cup \{0\}$
V'_{N+1}	Set of customers and recharging visits including depot instance $N+1$: $V'_{N+1} = V' \cup \{N+1\}$
$V'_{0,N+1}$	Set of customers and recharging visits including depot instances 0 and $N+1$: $V'_{0,N+1} = V' \cup \{0\} \cup \{N+1\}$
d_{ij}	Distance between vertices i and j
t_{ij}	Travel time between vertices i and j
C	Vehicle capacity
g	Recharging rate
h	Charge consumption rate
Q	Battery capacity
q_i	Demand of vertex i, 0 if $i \notin V$
e_i	Earliest start of service at vertex i
l_i	Latest start of service at vertex i
s_i	Service time at vertex i ($s_0, s_{N+1} = 0$)
τ_i	Decision variable specifying the time of arrival at vertex i
u_i	Decision variable specifying the remaining cargo on arrival at vertex i
y_i	Decision variable specifying the remaining battery capacity on arrival at vertex i
x_{ij}	Binary decision variable indicating if arc (i, j) is traveled

$$e_j \le \tau_j \le l_j \quad \forall j \in V'_{0,N+1}, \tag{7}$$

$$0 \le u_j \le u_i - q_i x_{ij} + C(1 - x_{ij}) \quad \forall i \in V'_0, \forall j \in V'_{N+1}, i \ne j, \tag{8}$$

$$0 \le u_0 \le C, \tag{9}$$

$$0 \le y_j \le y_i - (h \cdot d_{ij}) x_{ij} + Q(1 - x_{ij}) \quad \forall j \in V'_{N+1}, \forall i \in V, i \ne j, \tag{10}$$

$$0 \le y_j \le Q - (h \cdot d_{ij}) x_{ij} \quad \forall j \in V'_{N+1}, \forall i \in F'_0, i \ne j, \tag{11}$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in V'_0, j \in V'_{N+1}, i \ne j. \tag{12}$$

The objective of minimizing the traveled distance is defined in (1). Constraints (2) enforce the connectivity of customer visits and constraints (3) handle the connectivity of visits to recharging stations. Constraints (4) establish flow conservation by guaranteeing that at each vertex, the number of incoming arcs is equal to the number of outgoing arcs. Constraints (5) guarantee time feasibility for arcs leaving customers and the depot; constraints (6) do the same for arcs leaving recharging visits. As mentioned earlier, recharge times are for a complete recharge with rate g from the charge level y_i on arrival up to the maximum battery capacity Q. Constraints (7) enforce that every vertex is visited within its time window. Further, constraints (5)–(7) prevent the formation of subtours. Constraints (8) and (9) guarantee demand fulfillment at all customers by assuring a nonnegative cargo load upon arrival at any vertex including the depot. Finally, constraints (10) and (11) ensure that the battery charge never falls below 0.

4. A Hybrid VNS/TS Solution Method for the E-VRPTW

As a solution method for E-VRPTW, we use a combination of VNS and TS, a hybrid that has already proven its performance on routing and related problems (see, e.g., Melechovský, Prins, and Calvo 2005; Tarantilis, Zachariadis, and Kiranoudis 2008). VNS, proposed by Mladenović and Hansen (1997), is an effective metaheuristic performing local search on increasingly larger neighborhoods to efficiently explore the solution space and to avoid getting stuck in local optima. It has successfully been applied to a variety of combinatorial optimization problems, among them routing problems such as VRPTW with single or multiple depots (Bräysy 2003; Polacek et al. 2004).

TS is a powerful metaheuristic that guides local search heuristics to search a solution space economically and effectively (Glover and Laguna 1997). Starting from an initial solution, the best nontabu move is conducted at each iteration. To prevent the search heuristic from cycling, a memory structure called tabu list is used. TS methods have provided near-optimal solution qualities and proved their efficiency for many combinatorial optimization problems (Gendreau and Potvin 2010).

Figure 1 presents our solution method in pseudocode. After a preprocessing step removing infeasible arcs, we generate an initial solution S with a given number of vehicles as described in §4.1. Infeasible solutions are allowed during the search and evaluated based on a penalizing cost function (see §4.2). We first perform a feasibility phase during which the number of vehicles m is increased after no feasible solution has been found for a given number of η_{feas} iterations. After a feasible solution is found, another η_{dist} iterations are used to improve traveled distance. The search is guided by a VNS component described in §4.3. It uses the current VNS neighborhood $\mathcal{N}_\kappa$ to generate a random perturbation that serves as an initial solution for η_{tabu} iterations of the TS phase (§4.4). The acceptance criterion of the VNS is based on simulated annealing (SA).

4.1. Preprocessing and Generation of Initial Solution

As is commonly done, we apply a series of preprocessing steps to remove infeasible arcs (see, e.g.,

```
𝒩_κ ← set of VNS neighborhood structures for κ = 1, ..., κ_max
S ← generateInitialSolution()
κ ← 1
i ← 0
feasibilityPhase ← true
while feasibilityPhase ∨ (¬feasibilityPhase ∧ i < η_dist) do
  S′ ← generateRandomPoint(𝒩_κ(S))
  S″ ← applyTabuSearch(S′, η_tabu)
  if acceptSA(S″, S) then
    S ← S″
    κ ← 1
  else
    κ ← (κ mod κ_max) + 1
  end if
  if feasibilityPhase then
    if ¬ feasible(S) then
      if i = η_feas then
        addVehicle(S)
        i ← −1
      end if
    else
      feasibilityPhase ← false
      i ← −1
    end if
  end if
  i ← i + 1
end while
```

Figure 1 Overview of the VNS/TS Algorithm for Solving E-VRPTW

Psaraftis 1983; Savelsbergh 1985). An arc (v, w) can be removed from the set of possible arcs if one of the following conditions holds:

$$v, w \in V \wedge q_v + q_w > C, \tag{13}$$

$$v \in V_0', \; w \in V_{N+1}' \wedge e_v + s_v + t_{vw} > l_w, \tag{14}$$

$$v \in V_0', \; w \in V' \wedge e_v + s_v + t_{vw} + s_w + t_{wN+1} > l_0, \tag{15}$$

$$v, w \in V \wedge \forall j \in F_0', \; i \in F_{N+1}': \; h \cdot (d_{jv} + d_{vw} + d_{wi}) > Q. \tag{16}$$

Equations (13)–(15) are well-known preprocessing steps that are based on capacity and time window violations. Equation (16) is problem specific and refers to violations of the battery capacity. If the charge consumption of traveling an arc and traveling to and from that arc to any station or the depot is higher than the battery capacity, this arc can be labeled infeasible. Numerical studies showed that this preprocessing step is able to perceptibly reduce the number of feasible arcs on our E-VRPTW test instances.

We construct an initial solution similar to the approach proposed in Cordeau, Laporte, and Mercier (2001). First, all customers are sorted in increasing order of the angle between the depot, a randomly chosen point, and the customer. Then, customers are iteratively inserted into the active route at the position causing minimal increase in traveled distance until a violation of capacity or battery capacity occurs. If a violation occurs, we activate a new route until at most the predefined number of routes are opened; then the remaining customers are inserted into the last route. The battery capacity violation is determined under the assumption that no recharging possibility exists. To consider time window requirements, a customer u is only allowed to be inserted between successive vertices i, j if $e_i \le e_u \le e_j$. This rule helps to fulfill time windows, but feasibility is only guaranteed concerning capacity and battery capacity for all routes but the last.

4.2. Generalized Cost Function

VNS/TS allows infeasible solutions during the search process. We evaluate a solution by means of the following generalized cost function:

$$f_{\text{gen}}(S) = f(S) + \alpha P_{\text{cap}}(S) + \beta P_{\text{tw}}(S) + \gamma P_{\text{batt}}(S) + P_{\text{div}}(S), \tag{17}$$

where $f(S)$ denotes the total traveled distance, $P_{\text{cap}}(S)$ the total capacity violation, $P_{\text{tw}}(S)$ the time window violation, $P_{\text{batt}}(S)$ the battery capacity violation, $P_{\text{div}}(S)$ a diversification penalty, and α, β, and γ are factors weighting the violations. The penalty factors are initialized $(\alpha_0, \beta_0, \gamma_0)$ and dynamically updated between a given lower $(\alpha_{\min}, \beta_{\min}, \gamma_{\min})$ and upper bound $(\alpha_{\max}, \beta_{\max}, \gamma_{\max})$. To balance between diversification and intensification, they are increased by a factor δ after every η_{penalty} consecutive iterations in which the respective constraint has been violated and divided by δ if the respective constraint was satisfied.

In the following, we describe the efficient calculation of the constraint violations. We define a vehicle route r as a sequence of customers $\langle v_0, v_1, \ldots, v_n, v_{n+1}\rangle$, with v_0 and v_{n+1} representing the depot (cf. Nagata, Bräysy, and Dullaert 2010). A solution S to an E-VRPTW instance is defined as a set containing m routes, i.e., $S = \{r_k, k = 1, \ldots, m\}$. Vert$(r)$ denotes the set of vertices that are part of route r. Then, the capacity violation of a route r can be calculated as

$$P_{\text{cap}}(r) = \max\left(\sum_{v \in \text{Vert}(r)} q_v - C, 0\right).$$

The total capacity penalty of a solution S is calculated by adding the individual violations of all routes m:

$$P_{\text{cap}}(S) = \sum_{k=1}^{m} P_{\text{cap}}(r_k).$$

By saving forward and backward capacity requirements for each vertex (see, e.g., Kindervater and Savelsbergh 1997; Ibaraki et al. 2005), we are able to calculate the change in capacity violation in constant time $\mathcal{O}(1)$ for all neighborhood operators of our TS method, which are introduced in §4.4.

To calculate battery capacity violations, we define the following two variables for each vertex of a route $r = \langle v_0, v_1, \ldots, v_n, v_{n+1}\rangle$: $\Upsilon^{\rightarrow}_{v_i}$ contains the battery charge that is needed to travel either from the last recharging station visit or from the depot to vertex v_i, and $\Upsilon^{\leftarrow}_{v_i}$ is the battery charge that is needed to travel from v_i to either the next recharging station or the depot:

$$\Upsilon^{\rightarrow}_{v_i} = \begin{cases} h \cdot d_{v_{i-1}v_i} & \text{if } v_{i-1} \in F'_0, \\ \Upsilon^{\rightarrow}_{v_{i-1}} + h \cdot d_{v_{i-1}v_i} & \text{otherwise,} \end{cases} \quad i = 1, \ldots, n+1;$$

$$\Upsilon^{\leftarrow}_{v_i} = \begin{cases} h \cdot d_{v_i v_{i+1}} & \text{if } v_{i+1} \in F'_{n+1}, \\ \Upsilon^{\leftarrow}_{v_{i+1}} + h \cdot d_{v_i v_{i+1}} & \text{otherwise,} \end{cases} \quad i = 0, \ldots, n.$$

Using these variables, the battery capacity violation of a route r can be defined as the sum of individual violations at every visit to a recharging station and on return to the depot:

$$P_{\text{batt}}(r) = \sum_{v_i \in \text{Vert}(r) \cap F'_{N+1}} \max(\Upsilon^{\rightarrow}_{v_i} - Q, 0).$$

Thus, changes in battery capacity violation can be calculated in $\mathcal{O}(1)$ for all neighborhood operators described in §4.4 by calculating the change in battery capacity violation of the recharging stations immediately following the points of vertex insertion, removal, or merging of both routes.

To calculate time window violations, we adapt the time-window handling approach introduced in Nagata, Bräysy, and Dullaert (2010) and enhanced by Schneider, Sand, and Stenger (2013) to the E-VRPTW. The approach is based on the notion of time travel, i.e., the calculation of the violation at a customer that follows a customer with a time window violation is executed as if a travel back in time to the latest feasible arrival time at the preceding (violating) customer had taken place. By putting a penalty only on the first vertex where a time window is violated instead of propagating the violation along the entire route, the approach avoids penalizing good customer sequences only because they occur after a time window violation. Another important advantage of the approach is that potential time window violations for interroute moves can be calculated in $\mathcal{O}(1)$ for most classical neighborhood structures.

More precisely, for the VRPTW, by storing forward and backward time window penalty slacks, it is possible to calculate in constant time the time window penalties of a route $r_1 = \langle v_0, \ldots, u, w, \ldots, v_{n+1}\rangle$ that is constructed from two partial routes $\langle v_0, \ldots, u\rangle$ and $\langle w, \ldots, v_{n+1}\rangle$ or of a route $r_2 = \langle v_0, \ldots, u, v, w, \ldots, v_{n+1}\rangle$ that is constructed by inserting a vertex v between two partial routes $\langle v_0, \ldots, u\rangle$ and $\langle w, \ldots, v_{n+1}\rangle$.

This is not always possible if recharging stations are present because the recharging time at a station depends on the battery charge, which itself depends on the traveled distance to the recharging station. If the partial route $\langle w, \ldots, v_{n+1}\rangle$ contains a recharging station z, i.e., $\langle w, \ldots, z, z+1, \ldots, v_{n+1}\rangle$, slack variables have to be recalculated by traversing the partial route $\langle w, \ldots, z+1\rangle$ for r_1 and the partial route $\langle v, \ldots, z+1\rangle$ for r_2. Note that a recharging station in the first partial route $\langle v_0, \ldots, u\rangle$ or the vertex to be inserted v being a recharging station does not necessitate a recalculation.

4.3. The Variable Neighborhood Search Component

Within our hybrid VNS/TS heuristic, the VNS component is mainly used to diversify the search in a structured way. To explain the functionality of our VNS, we first briefly sketch a standard VNS procedure. Subsequently, we detail the specific characteristics of our implementation.

A general VNS algorithm works as follows: Given a predefined set of neighborhood structures and the current best solution S, VNS randomly generates a neighboring solution S' in the shaking phase by means of the neighborhood structure $\mathcal{N}_\kappa$. Next, a greedy local search is applied on S' to determine the local minimum S''. If S'' improves on the current best solution S, the VNS algorithm accepts the solution S'' and restarts with neighborhood $\mathcal{N}_1$ and the new starting solution S''. By contrast, if S'' is worse than the incumbent best solution, S'' is refused. In this case, VNS performs a random perturbation move according to the next more distant neighborhood structure $\mathcal{N}_{\kappa+1}$, starting again with S.

In our hybrid VNS/TS algorithm, the shaking phase is equal to the standard VNS approach, but the intensification phase and the acceptance criterion clearly differ. In the following, we detail the shaking phase, the local search phase, and the acceptance criterion used in the VNS component of our hybrid heuristic.

In every iteration, our VNS performs a random perturbation move according to the predefined neighborhood structure $\mathcal{N}_\kappa$. The neighborhood structures are all defined by means of the cyclic-exchange operator. In the cyclic-exchange, introduced by Thompson and Orlin (1989), customer sequences of arbitrary length are simultaneously transferred between routes. Figure 2 shows a cyclic-exchange involving three routes. Cyclic-exchange, or a variant that is restricted to two routes, called cross-exchange, is commonly used in the perturbation phase of VNS algorithms due to their strong diversification capabilities (see, e.g, Polacek et al. 2004; Hemmelmayr, Doerner, and Hartl 2009).

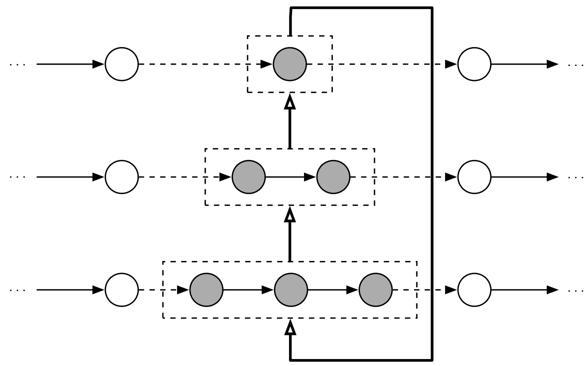

Figure 2 Example of a Cyclic-Exchange Move Involving Three Routes

Our κ neighborhood structures, shown in Table 2, are defined according to two parameters. The number of routes that form the cycle is equal to $\#Rts$. In each route r_k, we randomly select the number of successive vertices that form the translocation chain in the interval $[0, \min\{\Lambda_{\max}, n_k\}]$, where n_k denotes the number of customers and stations contained in r_k, and $\Lambda_{\max}$ denotes the maximum number of translocated vertices. The initial vertex of the translocation chain is randomly chosen for each route.

Contrary to the local descent commonly used in VNS approaches, we use a TS to improve the randomly generated solution S' in the intensification phase. The TS, which is detailed in §4.4, is run for η_{tabu} iterations. Note that the perturbation move is added to the tabu list to prevent its reversal. Subsequently, we compare the best solution S'' found during the TS to the initial solution S. Instead of accepting only improving solutions, we use an acceptance criterion based on the metaheuristic SA (Kirkpatrick, Gelatt, and Vecchi 1983) to further diversify the search. This method has been successfully applied in several VNS approaches, for example, in Hemmelmayr, Doerner, and Hartl (2009) and Stenger et al. (2013).

To be more precise, we always accept improving solutions, whereas deteriorating solutions are accepted according to the probability $e^{(-(f(S'')-f(S)))/T}$. At the beginning of the search, we initialize the temperature T to T_0 in a way that a solution value $f(S'')$, which is Δ_{SA} worse than $f(S)$, is accepted with a probability of 50%. In this way, deteriorating solutions are often accepted, which helps to diversify the search. After every VNS iteration, the temperature is linearly decreased with a cooling factor that is chosen such that the temperature is below 0.0001 during the last 20% of iterations. By continuously decreasing the temperature during the search, an intensification is achieved and, finally, only improving solutions are accepted.

Table 2 The κ-Neighborhood Structures Used in the VNS Defined by the Number of Involved Routes $\#Rts$ and the Maximum Number of Translocated Vertices $\Lambda_{\max}$

κ	$\#Rts$	$\Lambda_{\max}$	κ	$\#Rts$	$\Lambda_{\max}$	κ	$\#Rts$	$\Lambda_{\max}$
1	2	1	6	3	1	11	4	1
2	2	2	7	3	2	12	4	2
3	2	3	8	3	3	13	4	3
4	2	4	9	3	4	14	4	4
5	2	5	10	3	5	15	4	5

4.4. The Tabu Search Component

The TS phase starts from the solution S' generated by the perturbation move of the VNS component. In each iteration, the composite neighborhood $\mathcal{N}(S)$ of the TS is generated by applying the following neighborhood operators on every arc in the list of generator arcs (Toth and Vigo 2003): 2-opt*, relocate, exchange, and a new, problem-specific operator called *stationInRe*. Each move is evaluated, and the best nontabu move is performed. A move is superior if it is able to reduce the number of employed vehicles or if it has a lower cost function value calculated with Equation (17).

The 2-opt* operator is a modification of the 2-opt operator originally introduced in Lin (1965) and was specifically proposed for the VRPTW by Potvin and Rousseau (1995). It avoids the reversal of route directions by removing one arc from each route and reconnecting the first part of the first route with the second part of the second route and vice versa. We apply 2-opt* for interroute moves and define the operator for moving recharging stations, i.e., we allow the removal and insertion of arcs, including recharging stations. The relocate operator was introduced in Savelsbergh (1992) and removes one vertex from a route and inserts it into another route or a different position in the same route. Relocate is also defined for recharging stations and applied as intra- and interroute operator. The exchange operator, also introduced in Savelsbergh (1992), swaps the position of two vertices. The operator is applied for inter- and intraroute moves, but is not defined for recharging stations, i.e., we exclude the swapping of a recharging station with a customer or another station.

As the name suggests, the stationInRe operator performs insertions and removals of recharging stations. The operator is defined for all generator arcs (v, w), where v is a recharging station. Let w^- denote the predecessor of vertex w. If the arc (v, w) is not part of the current solution, stationInRe performs an insertion as depicted in Figure 3(a). If the arc is already present, a recharging station is removed as shown in Figure 3(b).

We set every arc ξ that is deleted from the solution by the execution of a move tabu, i.e., we forbid the reinsertion of the arc into specific parts of the solution for a specified number of iterations called

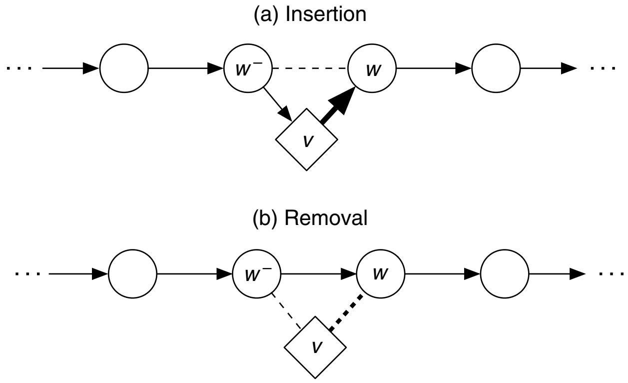

Figure 3 Insertion and Removal of a Recharging Station with the stationInRe Operator

Note. Generator arcs are shown in bold and removed arcs as dashed lines.

tabu tenure. Because station visits have a strong effect on charge levels and also on time windows due to the recharging times incurred, we define the tabu attribute (ξ, k, μ, ζ). It prohibits the insertion of arc ξ into route r_k between μ and ζ, where $\mu, \zeta \in F_{0,n+1}$ denote either a station or the depot. In this way, we allow the reinsertion of an arc into a different part of the route. The tabu tenure ϑ is randomly drawn from the interval $[\vartheta_{\min}, \vartheta_{\max}]$. The tabu status of a move is lifted if a so-called aspiration criterion is met—in our case, if a feasible new best solution is generated.

To further diversify the search, we adapt the continuous diversification mechanism presented in Cordeau, Laporte, and Mercier (2001) to E-VRPTW. To this end, we define vertex-based attributes (u, k, μ, ζ) to describe that customer/station u is positioned between stations/depot μ and ζ in route r_k. In this way, each solution S can be characterized by the attribute set $B(S) = \{(u, k, \mu, \zeta)\}$. For each attribute, the frequency $p_{uk\mu\zeta}$ of its addition to a solution in previous moves is memorized and used to penalize solutions according to the frequency of their attributes. Thus, we guide the search to explore the possibilities of using different stations and different positions of customers and stations (relative to other stations or the depot) within a route. A solution S' that deteriorates the current solution is penalized by:

$$P_{\mathrm{div}}(S') = \lambda_{\mathrm{div}} \cdot f(S') \cdot \sqrt{|V'|m} \sum_{(u,k,\mu,\zeta)\in B(S')} p_{uk\mu\zeta},$$

where λ_{div} denotes the diversification factor used to control the amount of diversification. The scaling factor $f(S') \cdot \sqrt{|V'|m}$ establishes a relation between the diversification penalty and the traveled distance and the investigated instance size ($|V'|$ customers and recharging visits and m vehicles). The TS procedure stops after η_{tabu} iterations.

5. Numerical Experiments

In this section, we present the extensive numerical testing conducted to evaluate the performance of our hybrid solution method. The first study evaluates the performance of VNS/TS on E-VRPTW instances. To be able to assess the solution quality, we use newly designed small instances that can be solved by means of the commercial solver CPLEX. In our second study, we analyze the efficiency of the algorithmic components of our hybrid heuristic, namely the VNS, TS, and SA, on a set of medium-sized E-VRPTW instances, which we design based on classical Solomon VRPTW instances. Finally, we demonstrate the strong performance concerning solution quality and runtime of our VNS/TS on available benchmark instances of the related problems MDVRPI, G-VRP, and VRPTW.

This section is structured as follows: After a discussion of the chosen parameter setting in §5.1, we describe the tests performed on E-VRPTW benchmark instances in §5.2, and those performed on benchmark instances of related problems in §5.3.

5.1. Experimental Environment and Parameter Setting

We performed all tests on a desktop computer equipped with an Intel Core i5 750 Processor clocked at 2.67 GHz with 4 GB RAM, running Windows 7 Professional. The VNS/TS is implemented as single-thread code in Java.

For our parameter studies, we used a randomly selected subset of 10 instances from the large 100-customer E-VRPTW instances described in §5.2.1. The iteration numbers of our VNS/TS were chosen to provide a good trade-off between solution quality and runtime from the authors' point of view. During the development of the method, we found that after a total of $\eta_{\mathrm{feas}} \cdot \eta_{\mathrm{tabu}} = 50{,}000$ iterations without finding a feasible solution on the E-VRPTW test instances, VNS/TS is very unlikely to find a feasible solution in later iterations. In a similar fashion, we observed that the largest improvements in distance minimization are achieved after $\eta_{\mathrm{dist}} \cdot \eta_{\mathrm{tabu}} = 20{,}000$ iterations, and afterward only marginal improvements occurred.

The ratio of VNS iterations to TS iterations and all other parameters were tuned in a systematic fashion following the approach described in Ropke and Pisinger (2006). We start from a base parameter setting that we found during the development of our VNS/TS and then tune the parameters consecutively, always keeping the best setting found for a parameter and proceeding with the next one. To keep the tuning effort low, we summarized parameters with similar functionality into groups that are assigned the same value (e.g., the start penalties α_0, β_0, γ_0 are treated as one parameter). For each parameter, we have tested only three values that were determined by varying

Table 3 Results Obtained on a Subset of the 100-Customer E-VRPTW Instances with Different Parameter Settings

	w_{SA}	4%	6%	**8%**
VNS	Δ_f (%)	0.34	0.52	0
Penalties				
	$\alpha_0, \beta_0, \gamma_0$	5.0	**10.0**	15.00
	Δ_f (%)	0.54	0	0.73
	$\alpha_{\min}, \beta_{\min}, \gamma_{\min}$	**0.5**	0.75	1.00
	Δ_f (%)	0	0.73	0.32
	$\alpha_{\max}, \beta_{\max}, \gamma_{\max}$	2,500	**5,000**	7,500
	Δ_f (%)	0.32	0	0.53
	δ	**1.2**	1.4	1.6
	Δ_f (%)	0	0.29	0.76
	$\eta_{penalty}$	1	**2**	3
	Δ_f (%)	0.82	0	0.57
TS				
	$\vartheta_{\min}$	**15**	20	25
	Δ_f (%)	0	0.69	0.41
	$\vartheta_{\max}$	20	**30**	40
	Δ_f (%)	0.24	0	0.40
	λ_{div}	**1.00**	1.25	1.50
	Δ_f (%)	0	0.13	0.41
	η_{tabu}	50	75	**100**
	Δ_f (%)	0.85	0.48	0

Notes. The best value for each parameter (group) is marked in bold and used as a final setting. Results are given as deviation of the best traveled distance achieved in 10 runs with the given setting to the result obtained with the final parameter setting (Δ_f). No deviation in the number of vehicles was found for any of the tested parameter values.

the base setting in both directions with a strength that seemed reasonable for the given parameter.

In this way, we conducted tests with different values for the SA deterioration percentage (Δ_{SA}), the initial penalties (α_0, β_0, γ_0), the penalty limits ($\alpha_{\min}, \beta_{\min}, \gamma_{\min}$ and $\alpha_{\max}, \beta_{\max}, \gamma_{\max}$), the penalty update factor (δ), the number of penalty update iterations (η_{penalty}), the minimal and maximal tabu tenure ($\vartheta_{\min}$ and $\vartheta_{\max}$), the diversification penalty (λ_{div}), and the number of TS iterations (η_{tabu}). For each parameter value, we conducted 10 runs and used the best solution found in these runs to assess the performance.

The results of the parameter tuning are shown in Table 3. Here, the middle of the three tested values represents the base setting, and the best value for each parameter (group) is marked in bold and used as a final setting for the studies. For all values, the percentage deviation Δ_f of the traveled distance to the best result found for this parameter is given. No deviation in the number of vehicles was found for any of the tested parameter values, and thus no gap is reported. We have conducted the test to determine the ratio of VNS and TS iterations for the feasibility phase, varying $\eta_{\text{tabu}} = 50$, 75, 100 while keeping $\eta_{\text{feas}} \cdot \eta_{\text{tabu}} = 50{,}000$. All settings achieved the same number of vehicles. Thus, the final setting for η_{tabu} is determined by the results of the distance minimization

Table 4 Overview of the Final Parameter Setting of VNS/TS Chosen for the Numerical Studies

VNS		Penalties		TS	
η_{feas}	500	$\alpha_0, \beta_0, \gamma_0$	10	$\vartheta_{\min}$	15
η_{dist}	200	$\alpha_{\min}, \beta_{\min}, \gamma_{\min}$	0.5	$\vartheta_{\max}$	30
Δ_{SA}	0.08	$\alpha_{\max}, \beta_{\max}, \gamma_{\max}$	5,000	λ_{div}	1.0
		δ	1.2	η_{tabu}	100
		η_{penalty}	2		

phase as shown in Table 3, where the total distance iterations were $\eta_{\text{tabu}} \cdot \eta_{\text{dist}} = 20{,}000$.

By tuning the parameters in the described fashion, the computational effort could be kept relatively moderate. Because all parameter settings achieved the same number of vehicles for the tested instances (which presents the primary objective of E-VRPTW), this shows that a deviation from the best parameter value found does not have a detrimental effect on the performance of our VNS/TS. This is further substantiated by the fact that all deviations of the traveled distance remain below 1%. Thus, it seems as if a number of reasonable parameter settings exist for our method, and finding one should be possible with acceptable effort. The determined parameter values are used for all of the following tests on E-VRPTW and on the related problems MDVRPI, G-VRP, and VRPTW. Table 4 summarizes the final setting.

5.2. Experiments on E-VRPTW Instances

Because we are the first to study E-VRPTW, no benchmark instances for assessing solution methods for this problem exist. We design two new sets of benchmark instances that we describe in §5.2.1. Section 5.2.2 presents the results of our studies on the generated instances.

5.2.1. Generation of E-VRPTW Benchmark Instances. We create two sets of benchmark instances for the E-VRPTW: a set of 56 large instances, each with 100 customers and 21 recharging stations, and a set of 36 small instances with 5, 10, and 15 customers per instance. All instances are created based on the benchmark instances for the VRPTW proposed by Solomon (1987). These instances are divided into three classes, depending on the geographical distribution of the customer locations: random customer distribution (R), clustered customer distribution (C), and a mixture of both (RC). Groups R1, C1, and RC1 have a short scheduling horizon, meaning that generally more vehicles are required to serve all customers than in R2, C2, and RC2, which have a long scheduling horizon. The instances within a group differ in terms of time window density and time window width. In the following, we detail the design of the large E-VRPTW instances based on the described VRPTW instances.

Given an original Solomon instance, we first determine the locations of the recharging stations. We locate one recharging station at the depot because a recharging possibility at the depot seems to be a reasonable claim. The location of the remaining 20 stations is determined in a random manner. However, we limit the possible locations to generate feasible and meaningful instances, i.e., every customer can be reached from the depot using at most two different recharging stations.

The battery capacity is set to the maximum of the following two values: (1) the charge needed to travel 60% of the average route length of the best-known solution to the corresponding VRPTW instance, and (2) twice the amount of battery charge required to travel the longest arc between a customer and a station. This procedure ensures that instances with geographically dispersed and remote customers stay feasible and, on the other hand, allows the formation of reasonable routes in instances with closely located customers. Furthermore, we thus guarantee that recharging stations have to be used. For the sake of simplicity, we set the consumption rate h to 1.0. The recharging rate g is set so that a complete recharge requires three times the average customer service time of the respective instance.

The detours for visits to recharging stations and the recharging times incurred make it impossible to comply with the customer time windows given in the original Solomon instances, i.e., some instances become infeasible because no possibility exists to reach certain customers within their original time window. Consequently, we generate new time windows to obtain feasible instances using a procedure very similar to the original one described in Solomon (1987). For a detailed description of our instance design, we refer the reader to the following URL: http://evrptw.wiwi.uni-frankfurt.de, where the generated instances are also available to be downloaded.

To generate the set of small instances, we start with the 56 large instances previously described. For each of the three sizes (5, 10, 15 customers), we randomly draw the respective number of customers from the large instances, thus generating 168 instances. The created instances are then solved with our VNS/TS heuristic and the solutions are inspected. For each group and instance size, we select the two instances whose solution uses the highest number of recharging stations. In this way, we create $6 \cdot 3 \cdot 2 = 36$ small test instances, which are denoted by the identifier of the underlying Solomon instance followed by the number of customers in the instance, e.g., R108-5.

5.2.2. Performance of VNS/TS on Small E-VRPTW Instances. We use the generated E-VRPTW test instances to analyze the performance of our VNS/TS heuristic on small E-VRPTW instances. To this end, we solve the instances with VNS/TS and compare the obtained results to the optimal (or near-optimal) solution found by the commercial solver CPLEX 12.2 using the E-VRPTW formulation presented in §3.

Table 5 **Comparison of Results Obtained with CPLEX and VNS/TS on the Small-Sized E-VRPTW Instances**

	CPLEX			VNS/TS			
Inst.	m	f	$t(s)$	m	f	$\Delta f(\%)$	$t(s)$
C101-5	2	257.75	81	2	257.75	0.00	0.21
C103-5	1	176.05	5	1	176.05	0.00	0.12
C206-5	1	242.55	518	1	242.55	0.00	0.14
C208-5	1	158.48	15	1	158.48	0.00	0.11
R104-5	2	136.69	1	2	136.69	0.00	0.13
R105-5	2	156.08	3	2	156.08	0.00	0.11
R202-5	1	128.78	1	1	128.78	0.00	0.11
R203-5	1	179.06	5	1	179.06	0.00	0.15
RC105-5	2	241.30	764	2	241.30	0.00	0.14
RC108-5	1	253.93	311	1	253.93	0.00	0.17
RC204-5	1	176.39	54	1	176.39	0.00	0.15
RC208-5	1	167.98	21	1	167.98	0.00	0.13
C101-10	3	393.76	171	3	393.76	0.00	0.77
C104-10	2	273.93	360	2	273.93	0.00	0.95
C202-10	1	304.06	300	1	304.06	0.00	0.71
C205-10	2	228.28	4	2	228.28	0.00	0.49
R102-10	3	249.19	389	3	249.19	0.00	0.65
R103-10	2	207.05	119	2	207.05	0.00	0.72
R201-10	1	241.51	177	1	241.51	0.00	0.78
R203-10	1	218.21	573	1	218.21	0.00	0.71
RC102-10	4	423.51	810	4	423.51	0.00	0.69
RC108-10	3	345.93	39	3	345.93	0.00	0.90
RC201-10	1	412.86	7,200	1	412.86	0.00	0.90
RC205-10	2	325.98	399	2	325.98	0.00	0.81
C103-15	3	384.29	7,200	3	384.29	0.00	15.37
C106-15	3	275.13	17	3	275.13	0.00	14.94
C202-15	2	383.62	7,200	2	383.61	0.00	13.41
C208-15	2	300.55	5,060	2	300.55	0.00	11.08
R102-15	5	413.93	7,200	5	413.93	0.00	19.55
R105-15	4	336.15	7,200	4	336.15	0.00	13.35
R202-15	2	358.00	7,200	2	358.00	0.00	13.17
R209-15	1	313.24	7,200	1	313.24	0.00	13.73
RC103-15	4	397.67	7,200	4	397.67	0.00	14.62
RC108-15	3	370.25	7,200	3	370.25	0.00	12.92
RC202-15	2	394.39	7,200	2	394.39	0.00	12.74
RC204-15	1	407.45	7,200	1	384.86	−5.87	15.57
Avg.			2,483.25			−0.16	5.03

Notes. m denotes the vehicle number and f the traveled distance. $t(s)$ denotes the total run-time in seconds. The maximum duration for CPLEX was set to two hours, so optimality is not guaranteed for CPLEX results that used the full time.

Table 5 provides an overview of the results. For both CPLEX and our heuristic we provide the computing time in seconds in column $t(s)$. For the solutions obtained with CPLEX, the vehicle number and traveled distance given in columns m and f correspond either to the optimal solution or the best upper bound found within 7,200 seconds. For VNS/TS, columns m and f provide the best solution found in 10 runs, and column Δf denotes the gap to the traveled distance found by CPLEX. No gap for the number of employed vehicles needs to be reported because the numbers obtained with CPLEX and VNS/TS are identical for all instances.

The results clearly show the ability of our VNS/TS heuristic to solve small E-VRPTW instances to optimality in only a few seconds. If CPLEX finds an optimum within 7,200 seconds, we always obtain the optimal solution independent of the instance structure or size. For most of the 15-customer instances and one 10-customer instance, CPLEX is not able to provide the optimal solution. On those instances, we either find a solution equal to the upper bound provided by CPLEX, or a better solution in one case. The results show that our VNS/TS is capable of determining highly efficient vehicle routes making use of the available recharging stations.

5.2.3. Analyzing the Effect of the VNS/TS Components. This section aims at demonstrating the positive effect achieved by the hybridization of VNS and TS. To this end, we compare the results obtained by our VNS/TS heuristic on the 100-customer E-VRPTW instances to the solutions found with (1) a VNS/TS heuristic that accepts only improving solutions after the TS phase instead of using an SA-based criterion (VNS/TS w/o SA), and (2) the pure TS heuristic.

An overview of the results is given in Table 6. For each heuristic, we provide the best solution found in 10 runs in columns m and f. Furthermore, we determine gaps to the best-known solution (BKS) found during the overall testing for both the number of vehicles (Δm) and the traveled distance (Δf). Finally, at the bottom of the table, the average computing time in minutes is reported in row t(min).

The results show that the VNS/TS heuristic performs best with a vehicle gap of 0 and an average gap of the traveled distance of 0.35% to the BKS. A comparison to the results obtained with VNS/TS w/o SA allows one to quantify the impact of the SA-based acceptance criterion. Using SA instead of simply accepting improving solutions reduces the cumulative number of vehicles (CNV) by three and yields a reduction of the traveled distance of 0.1% on average. Comparing the results of VNS/TS to those of the pure TS, we can see that the hybridization of VNS and TS is able to reduce the CNV by one. Concerning the traveled distance, VNS/TS reduces the gap to the BKS by more than half. Overall, the results show the positive effect of the hybridization of VNS, TS, and SA concerning the solution quality. Moreover, computing times can even be slightly reduced.

5.3. Performance on Benchmark Instances of Related Problems

The E-VRPTW is closely related to the MDVRPI and the G-VRP (see §2). For both problems, sets of benchmark instances exist. To demonstrate the performance of our VNS/TS heuristic on sets of large problems, we solve all benchmark instances available for the related problems and compare the results obtained to those reported for the competing algorithms, which were specifically designed for MDVRPI and G-VRP. Moreover, we present results of VNS/TS on the classical Solomon VRPTW benchmark instances and compare them to the state of the art.

5.3.1. Multidepot VRP with Interdepot Routes. For the MDVRPI (respectively the VRPIRF, see §2), two benchmark sets with a total of 76 instances are available from the literature. The first set was proposed by Crevier, Cordeau, and Laporte (2007) and includes 22 instances. The instances consist of 48–216 customers, 3–6 depots, and 4–6 vehicles. Depots are centered and customers are located in clusters. The second set was designed by Tarantilis, Zachariadis, and Kiranoudis (2008) and involves 54 instances. The set consists of 18 depot-customer combinations that were created following the design described in Crevier, Cordeau, and Laporte (2007) and involve 50–175 customers and 3–8 depots. From each of these 18 depot-customer combinations, three instances were created differing in the number of vehicles available.

In Table 7, we compare the results obtained with our VNS/TS on the instance set of Crevier, Cordeau, and Laporte (2007) to the solutions of the heuristics of Tarantilis, Zachariadis, and Kiranoudis 2008 (TZK) and Crevier, Cordeau, and Laporte 2007 (CCL). For CCL and our VNS/TS, we provide the best solution found in 10 runs (f_{best}) and the computing time in minutes (t(min)). By contrast, the value given in column f_{best} for TZK corresponds to the best solution ever found with the final parameter setting. We further provide the gap of the best (Δ_{best}) and average solution (Δ_{avg}) to the best-known solutions (BKS), which do not include the new best solutions that we found with our VNS/TS heuristic during the overall testing. They are shown in the last column ($\overline{\text{VNS/TS}}$), together with the percentage improvement in comparison to the formerly best known solutions.

Considering the complete set, our VNS/TS heuristic clearly outperforms the CCL approach in terms of solution quality and speed. We obtain an average gap to the best solution of 0.18% in about 27 minutes on average, whereas CCL achieves a 0.66% gap requiring more than double our computing time. In addition, we found 10 new overall best solutions during the testing. Tarantilis, Zachariadis, and Kiranoudis (2008) solved only the first subset of instances with their TZK approach. Compared to their results, we are on average 0.48% worse; however, a direct comparison is not adequate because the TZK results correspond to the best solution they ever found during their testing.

Table 8 compares the results of our VNS/TS heuristic with those of TZK on the instance set of

Table 6 Comparison of the Effect of Different Heuristic Components on the 100-Customer E-VRPTW Instances

	BKS		VNS/TS				VNS/TS w/o SA				TS			
Inst.	m	f	m	f	Δ_m	Δ_f(%)	m	f	Δ_m	Δ_f(%)	m	f	Δ_m	Δ_f(%)
c101	12	**1,053.83**	12	**1,053.83**	0	0.00	12	**1,053.83**	0	0.00	12	**1,053.83**	0	0.00
c102	11	**1,056.47**	11	1,057.16	0	0.07	11	**1,056.47**	0	0.00	11	1,069.35	0	1.22
c103	10	**1,041.55**	10	**1,041.55**	0	0.00	11	1,002.03	1	−3.79	10	1,134.36	0	8.91
c104	10	**979.51**	10	980.82	0	0.13	10	988.77	0	0.95	10	979.63	0	0.01
c105	11	**1,075.37**	11	**1,075.37**	0	0.00	11	**1,075.37**	0	0.00	11	1,079.69	0	0.40
c106	11	**1,057.87**	11	**1,057.87**	0	0.00	11	**1,057.87**	0	0.00	11	**1,057.87**	0	0.00
c107	11	**1,031.56**	11	**1,031.56**	0	0.00	11	**1,031.56**	0	0.00	11	1,033.08	0	0.15
c108	10	**1,100.32**	10	**1,100.32**	0	0.00	11	1,015.73	1	−7.69	11	1,015.73	1	−7.69
c109	10	**1,036.64**	10	1,051.84	0	1.47	10	**1,036.64**	0	0.00	10	1,051.36	0	1.42
c201	4	**645.16**	4	**645.16**	0	0.00	4	**645.16**	0	0.00	4	**645.16**	0	0.00
c202	4	**645.16**	4	**645.16**	0	0.00	4	**645.16**	0	0.00	4	**645.16**	0	0.00
c203	4	**644.98**	4	**644.98**	0	0.00	4	**644.98**	0	0.00	4	**644.98**	0	0.00
c204	4	**636.43**	4	**636.43**	0	0.00	4	**636.43**	0	0.00	4	**636.43**	0	0.00
c205	4	**641.13**	4	**641.13**	0	0.00	4	**641.13**	0	0.00	4	**641.13**	0	0.00
c206	4	**638.17**	4	**638.17**	0	0.00	4	**638.17**	0	0.00	4	**638.17**	0	0.00
c207	4	**638.17**	4	**638.17**	0	0.00	4	**638.17**	0	0.00	4	**638.17**	0	0.00
c208	4	**638.17**	4	**638.17**	0	0.00	4	**638.17**	0	0.00	4	**638.17**	0	0.00
r101	18	**1,670.8**	18	1,672.55	0	0.10	18	1,673.12	0	0.14	18	**1,670.8**	0	0.00
r102	16	**1,495.31**	16	1,535.81	0	2.71	16	1,522.84	0	1.84	16	**1,495.31**	0	0.00
r103	13	**1,299.17**	13	1,299.64	0	0.04	13	**1,299.17**	0	0.00	13	1,348.25	0	3.78
r104	11	**1,088.43**	11	**1,088.43**	0	0.00	11	1,143.69	0	5.08	11	1,097.09	0	0.80
r105	14	**1,461.25**	14	1,473.59	0	0.84	15	1,401.24	1	−4.11	14	1,514.36	0	3.63
r106	13	**1,344.66**	13	**1,344.66**	0	0.00	13	1,395.18	0	3.76	13	1,369.55	0	1.85
r107	12	**1,154.52**	12	**1,154.52**	0	0.00	12	1,158.13	0	0.31	12	1,162.9	0	0.73
r108	11	**1,050.04**	11	1,065.89	0	1.51	11	1,061.91	0	1.13	11	1,056.84	0	0.65
r109	12	**1,294.05**	12	**1,294.05**	0	0.00	12	1,341.01	0	3.63	12	1,308.62	0	1.13
r110	11	**1,126.74**	11	1,143.52	0	1.49	11	1,141.9	0	1.35	11	**1,126.74**	0	0.00
r111	12	**1,106.19**	12	1,124.06	0	1.62	12	1,107.52	0	0.12	12	1,123.96	0	1.61
r112	11	**1,026.52**	11	**1,026.52**	0	0.00	11	1,033.97	0	0.73	11	1,047.92	0	2.08
r201	3	**1,264.82**	3	**1,264.82**	0	0.00	3	**1,264.82**	0	0.00	3	1,266.26	0	0.11
r202	3	**1,052.32**	3	**1,052.32**	0	0.00	3	1,053.11	0	0.08	3	1,052.65	0	0.03
r203	3	**895.91**	3	912.86	0	1.89	3	914.68	0	2.10	3	914.1	0	2.03
r204	2	**790.57**	2	**790.57**	0	0.00	2	801.56	0	1.39	2	790.68	0	0.01
r205	3	**988.67**	3	**988.67**	0	0.00	3	1,000.96	0	1.24	3	997.15	0	0.86
r206	3	**925.2**	3	**925.2**	0	0.00	3	926.94	0	0.19	3	928.26	0	0.33
r207	2	**848.53**	2	852.73	0	0.49	2	**848.53**	0	0.00	2	855.99	0	0.88
r208	2	**736.6**	2	**736.6**	0	0.00	2	737.05	0	0.06	2	741.44	0	0.66
r209	3	**872.36**	3	**872.36**	0	0.00	3	877.4	0	0.58	3	874.74	0	0.27
r210	3	**847.06**	3	**847.06**	0	0.00	3	850.41	0	0.39	3	848.44	0	0.16
r211	2	**847.45**	2	866.21	0	2.21	2	860.32	0	1.52	2	861.17	0	1.62
rc101	16	**1,731.07**	16	**1,731.07**	0	0.00	16	1,766.44	0	2.04	16	1,753.35	0	1.29
rc102	15	**1,554.61**	15	**1,554.61**	0	0.00	15	1,556.08	0	0.09	15	1,559.95	0	0.34
rc103	13	**1,351.15**	13	1,353.55	0	0.18	13	**1,351.15**	0	0.00	13	1,355.36	0	0.31
rc104	11	**1,238.56**	11	1,249.23	0	0.86	11	1,267.55	0	2.34	11	1,280.82	0	3.41
rc105	14	**1,475.31**	14	1,483.38	0	0.55	14	**1,475.31**	0	0.00	14	1,479.56	0	0.29
rc106	13	**1,437.96**	13	1,440.19	0	0.15	13	1,469.99	0	2.23	13	**1,437.96**	0	0.00
rc107	12	**1,275.89**	12	**1,275.89**	0	0.00	12	1,280.44	0	0.36	12	1,284.47	0	0.67
rc108	11	**1,209.61**	11	1,238.81	0	2.41	11	1,227.88	0	1.51	11	**1,209.61**	0	0.00
rc201	4	**1,444.94**	4	1,447.2	0	0.16	4	**1,444.94**	0	0.00	4	1,446.03	0	0.08
rc202	3	**1,412.91**	3	**1,412.91**	0	0.00	3	**1,418.79**	0	0.42	3	1,425.17	0	0.87
rc203	3	**1,073.98**	3	1,078.28	0	0.40	3	1,077.16	0	0.30	3	1,084.66	0	0.99
rc204	3	**885.35**	3	889.22	0	0.44	3	886.03	0	0.08	3	889.22	0	0.44
rc205	3	**1,321.75**	3	**1,321.75**	0	0.00	3	1,353.54	0	2.41	3	1,360.39	0	2.92
rc206	3	**1,190.75**	3	1,191.13	0	0.03	3	1,204.93	0	1.19	3	1,207.77	0	1.43
rc207	3	**995.52**	3	**995.52**	0	0.00	3	1,015.6	0	2.02	3	1,010.66	0	1.52
rc208	3	**837.82**	3	838.03	0	0.03	3	838.41	0	0.07	3	838.03	0	0.03
Sum/Avg.					0	0.35			3	0.46			1	0.75
t(min)				15.34				16.22				16.01		

Notes. VNS/TS denotes the standard setting using an SA acceptance criterion. VNS/TS w/o SA denotes a combination of TS and a VNS only accepting improving solutions. TS denotes a pure TS without VNS shaking. Gaps are calculated to the best-known solution (BKS) found during the overall testing of the considered methods.

Table 7 Comparison of the Solutions Obtained by VNS/TS on the MDVRPI Instances Proposed by Crevier, Cordeau, and Laporte (2007) to Those of TZK and CCL

		CCL				TZK				VNS/TS				$\overline{\text{VNS/TS}}$	
Inst.	BKS	f_{best}	Δ_{best} (%)	Δ_{avg} (%)	t(min)	f^*_{best}	Δ^*_{best} (%)	Δ_{avg} (%)	t(min)	f_{best}	Δ_{best} (%)	Δ_{avg} (%)	t(min)	f	Δf (%)
a1	**1,179.79**	1,203.39	2.00	2.67	4.58	**1,179.79**	0.00	0.84	3.4	**1,179.79**	0.00	1.51	1.82	**1,179.79**	0.00
b1	**1,217.07**	**1,217.07**	0.00	1.28	9.17	**1,217.07**	0.00	0.66	7.8	**1,217.07**	0.00	0.80	7.14	**1,217.07**	0.00
c1	**1,883.05**	1,888.22	0.27	0.53	36.22	**1,883.05**	0.00	0.84	34.2	1,897.30	0.76	2.24	33.93	1,897.30	0.76
d1	**1,059.43**	**1,059.43**	0.00	1.59	8.55	**1,059.43**	0.00	0.46	5.9	1,060.10	0.06	0.34	1.82	**1,059.43**	0.00
e1	**1,309.12**	**1,309.12**	0.00	0.19	13.52	**1,309.12**	0.00	0.00	8.7	**1,309.12**	0.00	2.66	7.29	**1,309.12**	0.00
f1	**1,572.17**	1,592.25	1.28	1.87	41.41	**1,572.17**	0.00	0.87	38.8	1,584.06	0.76	3.03	34.61	1,575.57	0.22
g1	**1,181.13**	1,190.93	0.83	1.77	55.22	**1,181.13**	0.00	0.77	5.8	1,181.99	0.07	0.81	4.21	**1,181.13**	0.00
h1	**1,547.25**	1,566.75	1.26	3.31	32.07	**1,547.24**	0.00	1.96	11.1	1,566.19	1.22	2.27	18.03	1,562.56	0.99
i1	**1,925.99**	1,945.73	1.02	2.60	51.01	**1,925.99**	0.00	1.57	42.5	1,953.39	1.42	4.07	45.62	1,933.05	0.37
j1	1,117.20	1,144.41	2.44	3.99	58.90	1,117.20	0.00	1.04	5.5	**1,115.78**	−0.13	0.31	4.24	**1,115.78**	−0.13
k1	**1,580.39**	1,586.92	0.41	2.41	64.61	**1,580.39**	0.00	0.72	12.1	1,586.64	0.40	1.35	18.11	1,580.92	0.03
l1	**1,880.60**	1,897.74	0.91	1.94	104.27	**1,880.60**	0.00	1.27	51.4	1,902.72	1.18	2.78	46.14	1,894.05	0.72
Avg.			0.87	2.01	39.96		0.00	0.92	18.92		0.48	1.85	18.58		
a2	**997.94**	1,000.24	0.23	0.72	6.4					**997.94**	0.00	0.47	1.80	**997.94**	0.00
b2	1,307.28	1,307.28	0.00	1.98	14.7					1,301.21	−0.46	1.34	7.35	**1,291.19**	−1.23
c2	1,747.61	1,751.45	0.22	2.57	61.7					1,732.19	−0.88	0.76	18.05	**1,719.47**	−1.61
d2	1,871.42	1,877.03	0.30	1.43	40.5					1,892.62	1.13	1.97	35.10	**1,866.97**	−0.24
e2	1,942.85	1,974.13	1.61	2.72	73.8					1,940.52	−0.12	2.61	59.12	**1,928.06**	−0.76
f2	2,284.35	2,298.51	0.62	1.22	162.2					2,292.40	0.35	1.79	89.86	**2,275.28**	−0.40
g2	1,162.58	1,162.58	0.00	2.01	29.5					1,158.21	−0.38	−0.09	4.14	**1,152.92**	−0.83
h2	1,587.37	1,593.40	0.38	1.54	160.8					1,597.41	0.63	1.46	18.35	**1,580.55**	−0.43
i2	1,972.00	1,978.70	0.34	1.33	322.4					1,934.09	−1.92	−0.10	47.58	**1,925.52**	−2.36
j2	2,294.06	2,303.01	0.39	1.36	256.9					2,293.40	−0.03	1.58	91.30	**2,276.52**	−0.76
Avg.			0.41	1.69	112.89						−0.17	1.18	37.27		
Tot. Avg.			0.66	1.86	73.11						0.18	1.54	27.07		

Notes. BKS denotes the previously best-known solution. Gaps are calculated in dependence of BKS. Additionally, we provide the best solutions that we ever obtained on the instances during our testing activities in column $\overline{\text{VNS/TS}}$. t(min) denotes the average computational time of a run.
*This value corresponds to the best solution ever obtained with the final parameter setting.

Tarantilis, Zachariadis, and Kiranoudis (2008). On those instances, our VNS/TS heuristic shows a strong performance. Our results are on average 0.05% better than those of TZK. This is even more impressive when considering the fact that they provide only the best solution ever found. The gap of the average solution found by our VNS/TS is 1.44% and is hence also lower than the 1.6% gap of TZK. During our overall testing activities, we additionally obtained new best solutions for the majority of instances that improve the former ones by 0.45% on average.

5.3.2. Green VRP. The benchmark instances for the G-VRP were proposed in Erdogan and Miller-Hooks (2012) and consist of four sets, each involving 10 instances with 20 customers each. The instances differ in the customer distribution (random or clustered) and the number of available fuel stations (2 to 10). Furthermore, Erdogan and Miller-Hooks (2012) present a case study with 12 instances incorporating up to 500 customers.

Note that some customers contained in the test instances are infeasible, i.e., they cannot be served under the given restriction that each customer has to be reached in time with at most one stop at a station for refueling. Thus, these customers have to be identified and removed in a preprocessing step. Erdogan and Miller-Hooks (2012) report solutions found by their two heuristics MCWS and DBCA, as well as solutions determined with the commercial solver CPLEX. The CPLEX solution is, however, not the optimal solution to the instance. In their mathematical formulation, they fixed the number of vehicles to the value obtained with the best heuristic to get solutions that are comparable, i.e., to determine the best solution with a given number of vehicles.

We compiled an improved version of their G-VRP model, which is also available online at http://evrptw.wiwi.uni-frankfurt.de, and use it to solve the set of small instances with CPLEX 12.2. Instead of the hierarchical objective considered before, we minimize traveled distance for all tests on the G-VRP instances. In Table 9, we report the best upper bound for the traveled distance found by CPLEX in at most three hours of computing time in column f. In four cases, CPLEX was not able to determine any feasible solution. Furthermore, we solved all instances 10 times by means of our VNS/TS heuristic and report the best traveled

Table 8 Comparison of the Performance of Our VNS/TS Heuristic on the MDVRPI Instances Proposed by Tarantilis, Zachariadis, and Kiranoudis (2008) with the Solutions of TZK

		TZK				VNS/TS				$\overline{\text{VNS/TS}}$	
Instance	BKS	f^*_{best}	Δ^*_{best} (%)	Δ_{avg} (%)	t(min)	f_{best}	Δ_{best} (%)	Δ_{avg} (%)	t(min)	f_{best}	Δ_{best} (%)
50c3d2v	**2,209.83**	**2,209.83**	0.00	2.27	2.85	**2,209.83**	0.00	0.10	1.82	**2,209.83**	0.00
50c3d4v	**2,368.33**	**2,368.33**	0.00	2.18	2.23	**2,368.33**	0.00	0.89	1.84	**2,368.33**	0.00
50c3d6v	3,000.88	3,000.88	0.00	2.13	2.74	**2,999.29**	−0.05	0.75	1.88	**2,999.29**	−0.05
50c5d2v	**2,608.25**	**2,608.25**	0.00	2.87	1.54	**2,608.25**	0.00	1.01	2.00	**2,608.25**	0.00
50c5d4v	**3,086.58**	**3,086.58**	0.00	1.23	2.07	**3,086.58**	0.00	0.00	1.98	**3,086.58**	0.00
50c5d6v	3,552.00	3,552.00	0.00	0.90	3.04	3,552.00	0.00	0.28	2.02	**3,548.88**	−0.09
50c7d2v	**3,353.08**	**3,353.08**	0.00	2.37	3.16	3,353.83	0.02	3.10	2.38	**3,353.08**	0.00
50c7d4v	3,381.57	3,381.57	0.00	2.63	3.36	**3,380.27**	−0.04	0.54	2.10	**3,380.27**	−0.04
50c7d6v	4,097.80	4,097.80	0.00	0.25	3.42	**4,074.44**	−0.57	0.37	2.07	**4,074.44**	−0.57
75c3d2v	**2,678.80**	**2,678.80**	0.00	0.57	4.50	2,692.76	0.52	1.68	4.67	**2,678.80**	0.00
75c3d4v	**2,746.74**	**2,746.74**	0.00	1.95	3.38	**2,746.74**	0.00	0.17	4.33	**2,746.74**	0.00
75c3d6v	3,454.71	3,454.71	0.00	1.30	4.89	3,448.64	−0.18	0.41	4.34	**3,404.34**	−1.46
75c5d2v	**3,373.69**	**3,373.69**	0.00	2.98	3.29	3,386.64	0.38	2.34	5.09	**3,373.69**	0.00
75c5d4v	3,568.35	3,568.35	0.00	2.43	3.54	3,569.82	0.04	0.63	4.42	**3,553.46**	−0.42
75c5d6v	4,198.61	4,198.61	0.00	1.66	4.18	4,215.30	0.40	2.04	4.55	**4,193.86**	−0.11
75c7d2v	**3,569.02**	**3,569.02**	0.00	2.41	5.38	3,581.32	0.34	1.63	5.06	**3,569.02**	0.00
75c7d4v	3,830.43	3,830.43	0.00	2.13	5.51	3,830.43	0.00	1.70	4.61	**3,825.37**	−0.13
75c7d6v	**4,239.76**	**4,239.76**	0.00	2.02	4.29	4,244.35	0.11	0.75	4.80	4,242.08	0.05
100c3d3v	**3,123.51**	**3,123.51**	0.00	1.10	7.01	3,127.65	0.13	2.33	7.94	3,126.55	0.10
100c3d5v	3,552.50	3,552.50	0.00	2.37	7.31	3,548.75	−0.11	0.18	7.62	**3,548.44**	−0.11
100c3d7v	4,239.83	4,239.83	0.00	0.83	6.62	4,268.34	0.67	2.34	7.92	**4,239.50**	−0.01
100c5d3v	**4,053.95**	**4,053.95**	0.00	1.06	7.88	**4,053.95**	0.00	1.58	8.49	**4,053.95**	0.00
100c5d5v	**4,413.17**	**4,413.17**	0.00	2.69	7.20	4,424.81	0.26	5.52	7.70	4,415.48	0.05
100c5d7v	5,148.98	5,148.98	0.00	0.56	7.72	**5,142.52**	−0.13	0.15	7.93	**5,142.52**	−0.13
100c7d3v	**4,216.47**	**4,216.47**	0.00	0.61	8.53	4,242.38	0.61	1.62	8.87	**4,216.47**	0.00
100c7d5v	4,462.51	4,462.51	0.00	1.36	8.79	4,448.15	−0.32	0.67	8.00	**4,439.72**	−0.51
100c7d7v	4,897.47	4,897.47	0.00	1.55	8.35	4,916.62	0.39	3.83	8.10	**4,869.66**	−0.57
125c4d3v	3,920.05	3,920.05	0.00	1.18	8.73	3,966.61	1.19	3.62	13.23	**3,916.02**	−0.10
125c4d5v	4,315.68	4,315.68	0.00	1.30	9.00	**4,308.44**	−0.17	0.28	12.33	**4,308.44**	−0.17
125c4d7v	4,763.49	4,763.49	0.00	1.48	8.40	4,694.32	−1.45	0.57	12.54	**4,668.77**	−1.99
125c6d3v	**4,064.20**	**4,064.20**	0.00	0.78	9.19	4,117.41	1.31	3.40	13.56	4,076.04	0.29
125c6d5v	4,826.71	4,826.71	0.00	2.70	8.33	4,786.74	−0.83	0.22	13.09	**4,765.97**	−1.26
125c6d7v	5,325.28	5,325.28	0.00	2.65	9.18	5,221.52	−1.95	−0.55	12.89	**5,164.18**	−3.03
125c8d3v	4,553.28	4,553.28	0.00	2.73	10.23	4,574.82	0.47	1.51	14.98	**4,545.44**	−0.17
125c8d5v	5,045.65	5,045.65	0.00	1.38	9.64	**4,958.26**	−1.73	1.85	13.38	**4,958.26**	−1.73
125c8d7v	5,416.96	5,416.96	0.00	0.63	9.34	5,397.86	−0.35	1.05	13.38	**5,347.10**	−1.29
150c4d3v	**4,049.48**	**4,049.48**	0.00	0.01	9.71	4,072.95	0.58	3.05	21.84	4,069.72	0.50
150c4d5v	4,638.72	4,638.72	0.00	1.44	8.19	**4,622.77**	−0.34	0.61	19.11	**4,622.77**	−0.34
150c4d7v	5,176.50	5,176.50	0.00	1.30	8.00	5,163.02	−0.26	0.56	19.06	**5,137.69**	−0.75
150c6d3v	**4,057.09**	**4,057.09**	0.00	0.15	9.96	4,066.71	0.24	1.45	22.07	4,062.53	0.13
150c6d5v	**4,872.08**	**4,872.08**	0.00	0.54	10.23	4,931.13	1.21	2.42	21.16	4,876.91	0.10
150c6d7v	5,768.29	5,768.29	0.00	2.58	10.73	5,840.52	1.25	2.00	20.40	**5,712.01**	−0.98
150c8d3v	**4,653.90**	**4,653.90**	0.00	1.79	10.18	4,689.13	0.76	3.65	22.67	4,667.50	0.29
150c8d5v	5,113.77	5,113.77	0.00	1.10	11.62	5,116.55	0.05	1.69	19.60	**5,073.80**	−0.78
150c8d7v	5,665.23	5,665.23	0.00	0.00	12.01	5,648.32	−0.30	0.49	19.67	**5,612.02**	−0.94
175c4d4v	**4,706.76**	**4,706.76**	0.00	1.60	21.74	4,720.36	0.29	1.60	28.69	4,708.66	0.04
175c4d6v	**4,835.64**	**4,835.64**	0.00	2.58	23.01	4,863.88	0.58	2.50	26.71	4,841.51	0.12
175c4d8v	5,943.28	5,943.28	0.00	1.53	18.40	5,853.90	−1.50	−0.15	27.35	**5,832.26**	−1.87
175c6d4v	5,025.51	5,025.51	0.00	1.64	21.51	**5,011.01**	−0.29	1.90	29.28	**5,011.01**	−0.29
175c6d6v	5,431.34	5,431.34	0.00	0.11	22.54	5,382.57	−0.90	0.96	27.43	**5,360.35**	−1.31
175c6d8v	6,090.01	6,090.01	0.00	1.27	25.81	6,066.10	−0.39	1.08	27.97	**6,043.43**	−0.76
175c8d4v	5,878.58	5,878.58	0.00	2.59	24.90	5,840.25	−0.65	1.30	29.83	**5,822.55**	−0.95
175c8d6v	5,989.63	5,989.63	0.00	2.80	25.21	5,968.99	−0.34	2.24	27.78	**5,953.54**	−0.60
175c8d8v	6,943.63	6,943.63	0.00	1.90	26.70	6,840.04	−1.49	1.60	27.98	**6,775.68**	−2.42
Avg.			0.00	1.60	9.54		−0.05	1.44	12.79		−0.45

Notes. BKS denotes the previously best known solution. Gaps are calculated in dependence of BKS. Additionally, we provide the best solutions that we ever obtained on the instances during our testing activities in column $\overline{\text{VNS/TS}}$. t(min) denotes the average computational time of a run.

*This value corresponds to the best solution ever obtained with the final parameter setting.

Table 9 Results of VNS/TS on the Small-Sized G-VRP Instances

	CPLEX			MCWS				DBCA		VNS/TS				
	m	n	f	m	n	f	Δf (%)	f	Δf (%)	m	n	f	t(min)	Δf (%)
20c3sU1	**6**	20	**1,797.49**	**6**	20	1,818.35	1.16	1,797.51	0.00	**6**	20	**1,797.49**	0.69	0.00
20c3sU2	**6**	20	**1,574.77**	**6**	20	1,614.15	2.50	1,613.53	2.46	**6**	20	**1,574.77**	0.64	0.00
20c3sU3	**6**	20	**1,704.48**	7	20	1,969.64	15.56	1,964.57	15.26	**6**	20	**1,704.48**	0.64	0.00
20c3sU4	**5**	20	**1,482.00**	6	20	1,508.41	1.78	1,487.15	0.35	**5**	20	**1,482.00**	0.65	0.00
20c3sU5	6	20	**1,689.37**	**5**	20	1,752.73	3.75	1,752.73	3.75	6	20	**1,689.37**	0.67	0.00
20c3sU6	**6**	20	**1,618.65**	**6**	20	1,668.16	3.06	1,668.16	3.06	**6**	20	**1,618.65**	0.67	0.00
20c3sU7	**6**	20	**1,713.66**	**6**	20	1,730.45	0.98	1,730.45	0.98	**6**	20	**1,713.66**	0.64	0.00
20c3sU8	**6**	20	**1,706.50**	**6**	20	1,718.67	0.71	1,718.67	0.71	**6**	20	**1,706.50**	0.67	0.00
20c3sU9	**6**	20	**1,708.81**	**6**	20	1,714.43	0.33	1,714.43	0.33	**6**	20	**1,708.81**	0.66	0.00
20c3sU10	**4**	20	**1,181.31**	5	20	1,309.52	10.85	1,309.52	10.85	**4**	20	**1,181.31**	0.64	0.00
20c3sC1	**4**	20	**1,173.57**	5	20	1,300.62	10.83	1,300.62	10.83	**4**	20	**1,173.57**	0.62	0.00
20c3sC2	**5**	19	**1,539.97**	**5**	19	1,553.53	0.88	1,553.53	0.88	5	19	**1,539.97**	0.58	0.00
20c3sC3	**3**	12	**880.20**	4	12	1,083.12	23.05	1,083.12	23.05	**3**	12	**880.20**	0.25	0.00
20c3sC4	**4**	18	**1,059.35**	5	18	1,135.90	7.23	1,091.78	3.06	**4**	18	**1,059.35**	0.53	0.00
20c3sC5	7	19	—	**7**	19	2,190.68		2,190.68		7	19	**2,156.01**	0.60	
20c3sC6	**8**	17	**2,758.17**	9	17	2,883.71	4.55	2,883.71	4.55	**8**	17	**2,758.17**	0.71	0.00
20c3sC7	**4**	6	**1,393.99**	5	6	1,701.40	22.05	1,701.40	22.05	**4**	6	**1,393.99**	0.18	0.00
20c3sC8	**9**	18	**3,139.72**	10	18	3,319.74	5.73	3,319.74	5.73	**9**	18	**3,139.72**	0.62	0.00
20c3sC9	**6**	19	**1,799.94**	**6**	19	1,811.05	0.62	1,811.05	0.62	6	19	**1,799.94**	0.60	0.00
20c3sC10	**8**	15	—	**8**	15	2,648.84		2,644.11		8	15	**2,583.42**	0.45	
S1_2i6s	**6**	20	**1,578.12**	**6**	20	1,614.15	2.28	1,614.15	2.28	6	20	**1,578.12**	0.71	0.00
S1_4i6s	**5**	20	1,413.96	**5**	20	1,561.30	10.42	1,541.46	9.02	5	20	**1,397.27**	0.75	−1.18
S1_6i6s	**5**	20	**1,560.49**	6	20	1,616.20	3.57	1,616.20	3.57	5	20	**1,560.49**	0.73	0.00
S1_8i6s	**6**	20	**1,692.32**	**6**	20	1,902.51	12.42	1,882.54	11.24	6	20	**1,692.32**	0.74	0.00
S1_10i6s	**4**	20	**1,173.48**	5	20	1,309.52	11.59	1,309.52	11.59	**4**	20	**1,173.48**	0.71	0.00
S2_2i6s	**6**	20	**1,633.10**	**6**	20	1,645.80	0.78	1,645.80	0.78	6	20	**1,633.10**	0.75	0.00
S2_4i6s	**5**	19	1,555.20	6	19	**1,505.06**	−3.22	**1,505.06**	−3.22	5	19	1,532.96	0.88	−1.43
S2_6i6s	**7**	20	—	10	20	3,115.10		3,115.10		**7**	20	**2,431.33**	0.78	
S2_8i6s	**7**	16	**2,158.35**	9	16	2,722.55	26.14	2,722.55	26.14	**7**	16	**2,158.35**	0.57	0.00
S2_10i6s	**6**	*17*	—	**6**	16	1,995.62		1,995.62		6	*17*	**1,958.46**	0.61	
S1_4i2s	6	20	1,582.21	**6**	20	**1,582.20**	0.00	**1,582.20**	0.00	6	20	1,582.21	0.63	0.00
S1_4i4s	**5**	20	**1,460.09**	6	20	1,580.52	8.25	1,580.52	8.25	**5**	20	**1,460.09**	0.68	0.00
S1_4i6s	5	20	**1,397.27**	**5**	20	1,561.29	11.74	1,541.46	10.32	5	20	**1,397.27**	0.75	0.00
S1_4i8s	6	20	1,403.57	**6**	20	1,561.29	11.24	1,561.29	11.24	6	20	**1,397.27**	0.82	−0.45
S1_4i10s	5	20	1,397.27	**5**	20	1,536.04	9.93	1,529.73	9.48	5	20	**1,396.02**	0.85	−0.09
S2_4i2s	**4**	18	**1,059.35**	5	18	1,135.89	7.23	1,117.32	5.47	**4**	18	**1,059.35**	0.51	0.00
S2_4i4s	**5**	19	**1,446.08**	6	19	1,522.72	5.30	1,522.72	5.30	**5**	19	**1,446.08**	0.60	0.00
S2_4i6s	**5**	20	**1,434.14**	6	20	1,786.21	24.55	1,730.47	20.66	**5**	20	**1,434.14**	0.69	0.00
S2_4i8s	**5**	20	**1,434.14**	6	20	1,786.21	24.55	1,786.21	24.55	**5**	20	**1,434.14**	0.75	0.00
S2_4i10s	**5**	20	**1,434.13**	6	20	1,783.63	24.37	1,729.51	20.60	**5**	20	**1,434.13**	0.78	0.00
Avg.	5.58	18.8	1,575.98	6.13	18.78	1,781.42	8.52	1,774.15	7.94	5.58	18.8	1,645.45	0.65	−0.09

Notes. Comparison of the solutions obtained by the MCWS and DBCA heuristics, the solutions determined by our CPLEX implementation and those of our VNS/TS. f denotes the traveled distance and m the vehicle number of the best solution found by the respective method, and Δf the gap to the CPLEX solution. t(min) reports the average computing time in minutes. We terminate CPLEX after three hours, so optimality is not guaranteed for any of the reported CPLEX solutions. Numbers in bold indicate the best solution found. Note that in one case, we identified a higher number of feasible customers n (number in italics) than Erdogan and Miller-Hooks (2012).

distance f and the average computing time in minutes (t(min)).

We compare our results to those obtained with the MCWS and DBCA heuristics, for which the best solution obtained in multiple runs is reported (no exact number of runs is given in the paper). Unfortunately, no computing times are available for these heuristics. In Table 9, we further report the gap of the best solution found by each of the heuristic methods to the CPLEX solution (Δf). Column n provides the number of feasible customers and m the number of vehicles required in the respective solutions.

Our VNS/TS heuristic clearly outperforms both heuristic methods proposed by Erdogan and Miller-Hooks (2012), which both achieve an average gap to the CPLEX solution of about 8%. On all instances, VNS/TS finds the best CPLEX solution or even a solution that improves on the upper bound, resulting in an average gap of −0.09%. It is also worth mentioning that, compared to MCWS and DBCA, the VNS/TS

Table 10 Results on the Large-Scale G-VRP Instances

			MCWS		DBCA		VNS/TS			
	m	n	f	Δf (%)	f	Δf (%)	m	n	f	t(min)
111c_21	20	109	5,626.64	17.29	5,626.64	17.29	**17**	109	**4,797.15**	21.76
111c_22	20	109	5,610.57	16.83			**17**	109	**4,802.16**	23.56
111c_24	20	109	5,412.48	13.07			**17**	109	**4,786.96**	21.90
111c_26	20	109	5,408.38	13.18			**17**	109	**4,778.62**	25.12
111c_28	20	109	5,331.93	11.10			**17**	109	**4,799.15**	24.17
200c	35	190	10,428.59	16.35	10,413.59	16.18	**35**	*192*	**8,963.46**	76.65
250c	41	235	11,886.61	10.06	11,886.61	10.06	**39**	*237*	**10,800.18**	120.90
300c	49	281	14,242.56	13.08	14,229.92	12.98	**46**	*283*	**12,594.77**	182.23
350c	57	329	16,471.10	15.00	16,460.30	14.92	**51**	329	**14,323.02**	232.03
400c	67	378	19,472.10	15.56	19,099.04	13.35	**61**	378	**16,850.21**	305.12
450c	75	424	21,854.17	18.00	21,854.19	18.00	**68**	424	**18,521.23**	525.52
500c	84	471	24,527.46	15.85	24,517.08	15.81	**76**	471	**21,170.90**	356.01
Average	42.33	237.75	12,189.38	14.61	15,510.92	14.82	38.42	238.25	10,598.98	159.58

Notes. Comparison of the solutions obtained by the MCWS and DBCA heuristics and those of our VNS/TS. For VNS/TS, f denotes the best solution found in 10 runs and t(min) reports the average computing time in minutes. Numbers in bold indicate the best solution found. As these are all provided by VNS/TS, we give the percentage gap to the VNS/TS solution for MCWS and DBCA in column Δf. Note that in some cases, our preprocessing identified a higher number of feasible customers n (numbers in italics) than Erdogan and Miller-Hooks (2012).

solutions reduce the number of vehicles in almost half of the instances, while requiring less than 40 seconds of computing time on average.

We additionally solve all large instances of the case study presented by Erdogan and Miller-Hooks (2012). In Table 10, we compare the results obtained with our heuristic to those of MCWS and DBCA. As VNS/TS provides the best solution for all instances, the value given in column Δf denotes the gap to the best solution found by VNS/TS in 10 runs. The results obtained by our heuristic on the set of large instances is quite convincing. We reduce the traveled distance of the solutions of MCWS and DBCA by almost 15%. In addition, we require significantly fewer vehicles on average.

To conclude, although our approach is not specifically tailored to the MDVRPI or G-VRP, we are able to outperform the state-of-the-art heuristics on the G-VRP and the second MDVRPI benchmark set. On the first MDVRPI set, we obtain competitive results while requiring moderate computing times.

5.3.3. VRP with Time Windows. Finally, we conduct tests on the Solomon VRPTW instances to further assess the quality of our VNS/TS. The classical VRPTW represents a special case of E-VRPTW adequate for scenarios in which no recharges are necessary. Such problems are of practical interest if service areas are small and/or customer density is high because then the battery capacity may be sufficient to serve all routes without recharging. Such application scenarios are likely to become more frequent if the maximum driving ranges of ECVs continue to grow in rapid fashion.

As commonly done for heuristic VRPTW solution methods (see, e.g., Taillard et al. 1997; Cordeau, Laporte, and Mercier 2001; Ibaraki et al. 2008), we initially set the number of routes to the best-known solution from the literature. During our tests, we recognized that VNS/TS is able to find a feasible solution with the given vehicle number in significantly less than 500 VNS iterations for the Solomon instances. Therefore, we reduced the number of feasibility iterations to $\eta_{\text{feas}} = 300$ and the number of distance minimization iterations to $\eta_{\text{dist}} = 50$ to increase the competitiveness of VNS/TS runtimes. All other parameters remain set as described in §5.1.

In Table 11, we present the results of VNS/TS in comparison to the currently best-performing VRPTW method, the memetic algorithm of Nagata, Bräysy, and Dullaert (2010) (reported as NBD), and the best-known solutions in terms of vehicle number m and traveled distance f, which we collected from Nagata, Bräysy, and Dullaert (2010) and the SINTEF website

Table 11 The Results of VNS/TS on the Solomon Benchmark Set Concerning the Number of Vehicles m and the Traveled Distance f in Comparison to the Best VRPTW Method from the Literature (NBD) and the Best-Known Solutions (BKS)

Problem group	BKS (m | f)	NBD (m | f)	VNS/TS (m | f)
R1	11.92 | 1,209.89	11.92 | 1,210.34	11.92 | 1, 217.71
R2	2.73 | 951.05	2.73 | 951.03	2.73 | 965.98
C1	10.0 | 828.38	10.0 | 828.38	10.0 | 828.38
C2	3.0 | 589.86	3.0 | 589.86	3.0 | 589.86
RC1	11.5 | 1,384.16	11.5 | 1,384.16	11.5 | 1,395.04
RC2	3.25 | 1,119.17	3.25 | 1,119.24	3.25 | 1,140.82
CNV	405	405	405
CTD	57,181	57,187	57,700

Note. Average results for each problem group, the cumulative number of vehicles (CNV), and the cumulative traveled distance (CTD) are provided.

at http://www.sintef.no/Projectweb/TOP/VRPTW/Solomon-benchmark/100-customers. We follow the common procedure and give averages over the problem groups R1, R2, C1, C2, RC1, and RC2, the cumulative number of vehicles (CNV), and the cumulative traveled distance (CTD). For VNS/TS, we report the best solution found in 10 runs. For the memetic algorithm of Nagata, Bräysy, and Dullaert (2010), the best solution of five runs is given.

Concerning the solution quality, VNS/TS achieves a gap to the CNV of the BKS of 0% and a gap to the CTD of the BKS of 0.91%. Moreover, VNS/TS has an average computational time of 124 seconds per instance compared to an average runtime of 300 seconds of NBD. Thus, the performance of VNS/TS is comparable with the most successful VRPTW methods because a CNV of 405 is only achieved by relatively few of the numerous methods proposed for VRPTW. This result is even more persuasive, considering that VNS/TS was designed to solve E-VRPTW instances and is thus more flexible than specialized VRPTW methods. Further, the parameter tuning was carried out using E-VRPTW instances.

6. Conclusion

In this paper, we present a new vehicle routing problem for determining cost-optimal routes for ECVs. The E-VRPTW considers a limited battery capacity and the possibility of en route recharging at one of the available stations with a recharging time that depends on the battery level on arrival at the station. Furthermore, customer time windows and vehicle capacity constraints are incorporated into the E-VRPTW model to represent real-world requirements.

We develop a hybrid VNS/TS heuristic that makes use of the strong diversification effect of VNS and involves a TS heuristic to efficiently search the solution space from a randomly generated solution of the VNS component. In numerical studies performed on newly designed E-VRPTW benchmark instances, we demonstrate the strong performance of our metaheuristic and the positive effect of combining the two metaheuristics VNS and TS. The quality of our method is further substantiated by very convincing results on test instances of the related problems MDVRPI, G-VRP, and VRPTW.

Thus, our method seems able to successfully assist routing and recharging decisions for ECVs employed in real-world delivery operations. In this way, we provide a first step toward the efficient and effective routing of ECVs, which is one of the prerequisites for rendering ECVs competitive to conventional vehicles. This shall facilitate the introduction and profitable use of ECVs in the transportation market and thus support the spreading of green logistics practices.

Future work should enhance the accuracy of the routing model by considering payloads, grades, and vehicle speeds as described in §3. The integration of the energy consumption model of Davis and Figliozzi (2013) could be a first step in this direction. Moreover, hard time window restrictions can strongly increase the number of vehicles necessary to fulfill all requirements. This motivates the investigation of E-VRPTW with soft time windows because ECVs have very high acquisition costs.

References

Artmeier A, Haselmayr J, Leucker M, Sachenbacher M (2010) The shortest path problem revisited: Optimal routing for electric vehicles. *KI 2010: Advances in Artificial Intelligence*, Lecture Notes in Computer Science, Vol. 6359 (Springer, Berlin), 309–316.

Baldacci R, Mingozzi A, Roberti R (2012) Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *Eur. J. Oper. Res.* 218(1):1–6.

Bektaş T, Laporte G (2011) The pollution-routing problem. *Transportation Res. Part B* 45(8):1232–1250.

Beliën J, De Boeck L, Van Ackere J (2014) Municipal solid waste collection and management problems: A literature review. *Transportation Sci.* 48(1):78–102.

Beltran B, Carrese S, Cipriani E, Petrelli M (2009) Transit network design with allocation of green vehicles: A genetic algorithm approach. *Transportation Res. Part C* 17(5):475–483.

Boostani A, Ghodsi R, Miab AK (2010) Optimal location of compressed natural gas (CNG) refueling station using the arc demand coverage model. *Proc. 2010 Fourth Asia Internat. Conf. Math./Analytical Model. Comput. Simulation, Kota Kinabalu, Malaysia*, 193–198.

Botsford C, Szczepanek A (2009) Fast charging vs. slow charging: Pros and cons for the new age of electric vehicles. *EVS24 Internat. Battery, Hybrid and Fuel Cell Electric Vehicle Sympos., Stavanger, Norway*.

Bräysy O (2003) A reactive variable neighborhood search for the vehicle-routing problem with time windows. *INFORMS J. Comput.* 15(4):347–368.

Bräysy O, Gendreau M (2005) Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. *Transportation Sci.* 39(1):104–118.

Conrad RG, Figliozzi MA (2011) The recharging vehicle routing problem. *Proc. 2011 Indust. Engrg. Res. Conf., Reno, NV*.

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.

Crevier B, Cordeau J-F, Laporte G (2007) The multi-depot vehicle routing problem with inter-depot routes. *Eur. J. Oper. Res.* 176(2):756–773.

Davis BA, Figliozzi MA (2013) A methodology to evaluate the competitiveness of electric delivery trucks. *Transportation Res. Part E* 49(1):8–23.

Dekker R, Bloemhof J, Mallidis I (2012) Operations research for green logistics: An overview of aspects, issues, contributions and challenges. *Eur. J. Oper. Res.* 219(3):671–679.

Demir E, Bektaş T, Laporte G (2012) An adaptive large neighborhood search heuristic for the pollution-routing problem. *Eur. J. Oper. Res.* 223(2):346–359.

EC (2011a) European Commission: Reducing emissions from transport. Accessed February 7, 2014, http://ec.europa.eu/clima/policies/transport/index_en.htm.

EC (2011b) European Commission: White Paper—Roadmap to a single European transport area—Towards a competitive and resource efficient transport system. Accessed February 7, 2014, http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2011:0144:FIN:EN:PDF.

EEA (2012) European Environment Agency: Greenhouse gas data—Emissions share by sector in EU27, (2010). Accessed February 7, 2014, http://www.eea.europa.eu/data-and-maps/data/data-viewers/greenhouse-gases-viewer.

EPEC (2011) European Parliament and European Council: Regulation (EU) No 510/2011—Setting emission performance standards for new light commercial vehicles as part of the union's integrated approach to reduce CO_2 emissions from light-duty vehicles. Accessed February 7, 2014, http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=0J:L:2011:145:0001:0018UR:EN:PDF.

Erdogan S, Miller-Hooks E (2012) A green vehicle routing problem. *Transportation Res. Part E* 48(1):100–114.

Faulin J, Lera-López F, Juan AA (2011) Optimizing routes with safety and environmental criteria in transportation management in Spain: A case study. *Internat. J. Inform. Systems Supply Chain Management* 4(3):38–59.

Feng W, Figliozzi M (2013) An economic and technological analysis of the key factors affecting the competitiveness of electric commercial vehicles: A case study from the USA market. *Transportation Res. Part C* 26:135–145.

Figliozzi MA (2010) Vehicle routing problem for emissions minimization. *Transportation Res. Record: J. Transportation Res. Board* 2197:1–7.

Figliozzi MA (2011) The impacts of congestion on time-definitive urban freight distribution networks CO_2 emission levels: Results from a case study in Portland, Oregon. *Transportation Res. Part C* 19(5):766–778.

Gendreau M, Potvin J-Y (2010) Tabu search. Gendreau M, Potvin J-Y, eds. *Handbook of Metaheuristics, International Series in Operations Research & Management Science*, Vol. 146 (Springer, New York), 41–59.

Gendreau, M, Tarantilis CD (2010) Solving large-scale vehicle routing problems with time windows: The state-of-the-art. Technical report 2010-04, CIRRELT, Montréal, https://www.cirrelt.ca/DocumentsTravail/CIRRELT-2010-04.pdf.

Glover F, Laguna M (1997) *Tabu Search* (Springer, Boston).

Gonçalves F, Cardoso SR, Relvas S, Barbosa-Póvoa APFD (2011) Optimization of a distribution network using electric vehicles: A VRP problem. *IO2011—15° Congresso da associação Portuguesa de Investigação Operacional* (INESC, Lisboa, Portugal).

Guzzella L, Sciarretta A (2005) *Vehicle Propulsion Systems: Introduction to Modeling and Optimization* (Springer, Berlin, Heidelberg).

He F, Wu D, Yin Y, Guan Y (2013) Optimal deployment of public charging stations for plug-in hybrid electric vehicles. *Transportation Res. Part B* 47:87–101.

Hemmelmayr VC, Doerner KF, Hartl RF (2009) A variable neighborhood search heuristic for periodic routing problems. *Eur. J. Oper. Res.* 195(2):791–802.

Ibaraki T, Imahori S, Kubo M, Masuda T, Uno T, Yagiura M (2005) Effective local search algorithms for routing and scheduling problems with general time-window constraints. *Transportation Sci.* 39(2):206–232.

Ibaraki T, Imahori S, Nonobe K, Sobue K, Uno T, Yagiura M (2008) An iterated local search algorithm for the vehicle routing problem with convex time penalty functions. *Discrete Appl. Math.* 156(11):2050–2069.

Ichimori T, Ishii H (1981) Routing a vehicle with the limitation of fuel. *J. Oper. Res. Soc. Japan* 24(3):277–281.

Ichimori T, Ishii H, Nishida T (1983) Two routing problems with the limitation of fuel. *Discrete Appl. Math.* 6(1):85–89.

IPC (2011) International Post Corporation: Green Flash. Accessed February 7, 2014, http://www.ipc.be/~/media/Documents/PUBLIC/market-flash/Green%20Issues/GF18.pdf.

Jabali O, Van Woensel T, de Kok AG (2012) Analysis of travel times and CO_2 emissions in time-dependent vehicle routing. *Production Oper. Management* 21(6):1060–1074.

Kek AGH, Cheu RL, Meng Q (2008) Distance-constrained capacitated vehicle routing problems with flexible assignment of start and end depots. *Math. Comput. Model.* 47(1–2):140–152.

Kindervater G, Savelsbergh M (1997) Vehicle routing: Handling edge exchanges. Aarts E, Lenstra JK, eds. *Local Search in Combinatorial Optimization* (John Wiley & Sons, Hoboken, NJ), 337–360.

Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. *Science* 220(4598):671–680.

Kleindorfer PR, Singhal K, Wassenhove LN (2005) Sustainable operations management. *Production Oper. Management* 14(4): 482–492.

Kleindorfer PR, Neboian A, Roset A, Spinler S (2012) Fleet renewal with electric vehicles at La Poste. *Interfaces* 42(5):465–477.

Laporte G, Nobert Y, Desrochers M (1985) Optimal routing under capacity and distance restrictions. *Oper. Res.* 33(5):1050–1073.

Li C-L, Simchi-Levi D, Desrochers M (1992) On the distance constrained vehicle routing problem. *Oper. Res.* 40(4):790–799.

Lin S (1965) Computer solutions of the traveling salesman problem. *Bell System Tech. J.* 44(10):2245–2269.

Maden W, Eglese R, Black D (2010) Vehicle routing and scheduling with time-varying data: A case study. *J. Oper. Res. Soc.* 61(3):515–522.

Mak H-Y, Rong Y, Shen Z-JM (2013) Infrastructure planning for electric vehicles with battery swapping. *Management Sci.* 59(7):1557–1575.

Marra F, Yang GY, Traholt C, Larsen E, Rasmussen CN, Shi Y (2012) Demand profile study of battery electric vehicle under different charging options. *IEEE Power and Energy Soc. General Meeting, San Diego*, 1–7.

McKinnon A, Cullinane S, Whiteing A, Browne M, eds. (2010) *Green Logistics: Improving the Environmental Sustainability of Logistics* (Kogan Page, London).

Mehrez A, Stern HI (1985) Optimal refueling strategies for a mixed-vehicle fleet. *Naval Res. Logist. Quart.* 32(2):315–328.

Melechovský J, Prins C, Calvo R (2005) A metaheuristic to solve a location-routing problem with non-linear costs. *J. Heuristics* 11(5–6):375–391.

Melkman AA, Stern HI, Mehrez A (1986) Optimal refueling sequence for a mixed fleet with limited refuelings. *Naval Res. Logist. Quart.* 33(4):759–762.

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

Nagata Y, Bräysy O, Dullaert W (2010) A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. *Comput. Oper. Res.* 37(4):724–737.

Polacek M, Hartl RF, Doerner K, Reimann M (2004) A variable neighborhood search for the multi depot vehicle routing problem with time windows. *J. Heuristics* 10(6):613–627.

Potvin J-Y, Rousseau J-M (1995) An exchange heuristic for routing problems with time windows. *J. Oper. Res. Soc.* 46(12): 1433–1446.

Psaraftis HN (1983) *k*-interchange procedures for local search in a precedence-constrained routing problem. *Eur. J. Oper. Res.* 13(4):391–402.

Rochat Y, Taillard ED (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. *Transportation Sci.* 40(4):455–472.

Savelsbergh MWP (1985) Local search in routing problems with time windows. *Ann. Oper. Res.* 4(1):285–305.

Savelsbergh MWP (1992) The vehicle routing problem with time windows: Minimizing route duration. *ORSA J. Comput.* 4(2):146–154.

Sbihi A, Eglese RW (2010) Combinatorial optimization and green logistics. *Ann. Oper. Res.* 175(1):159–175.

Schneider M, Sand B, Stenger A (2013) A note on the time travel approach for handling time windows in vehicle routing problems. *Comput. Oper. Res.* 40(10):2564–2568.

Solomon MM (1987) Algorithms for the vehicle routing and scheduling problems with time window constraints. *Oper. Res.* 35(2):254–265.

Srivastava SK (2007) Green supply-chain management: A state-of-the-art literature review. *Internat. J. Management Rev.* 9(1): 53–80.

Stenger A, Vigo D, Enz S, Schwind M (2013) An adaptive variable neighborhood search algorithm for a vehicle routing problem arising in small package shipping. *Transportation Sci.* 47(1): 64–80.

Taillard É, Badeau P, Gendreau M, Guertin F, Potvin J-Y (1997) A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Sci.* 31(2):170–186.

Tarantilis CD, Zachariadis EE, Kiranoudis CT (2008) A hybrid guided local search for the vehicle-routing problem with intermediate replenishment facilities. *INFORMS J. Comput.* 20(1):154–168.

Thompson PM, Orlin JB (1989) Theory of cyclic transfers. Working paper, Operations Research Center, MIT, Cambridge, MA.

Toth P, Vigo D, eds. (2002) *The Vehicle Routing Problem*, Monographs on Discrete Mathematics and Applications (SIAM, Philadelphia).

Toth P, Vigo D (2003) The granular tabu search and its application to the vehicle-routing problem. *INFORMS J. Comput.* 15(4): 333–346.

Transport for London (2013) Source London. Accessed February 7, 2014, https://www.sourcelondon.net/.

Tredeau FP, Salameh ZM (2009) Evaluation of lithium iron phosphate batteries for electric vehicles application. *IEEE Vehicle Power and Propulsion Conf., Dearborn, MI*, 1266–1270.

Ubeda S, Arcelus FJ, Faulin J (2011) Green logistics at Eroski: A case study. *Internat. J. Production Econom.* 131(1):44–51.

Wang H, Shen J (2007) Heuristic approaches for solving transit vehicle scheduling problem with route and fueling time constraints. *Appl. Math. Comput.* 190(2):1237–1249.

Wang Y-W, Lin C-C (2009) Locating road-vehicle refueling stations. *Transportation Res. Part E* 45(5):821–829.

Wang Y-W, Wang C-R (2010) Locating passenger vehicle refueling stations. *Transportation Res. Part E* 46(5):791–801.

Zhang X, Mi C (2011) *Vehicle Power Management—Modeling, Control and Optimization* (Springer, London).