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

**Michael Schneider, Andreas Stenger, Dominik Goeke**

Technical Report 02/2012

Michael Schneider
Chair of Business Information Systems and Operations Research
University of Kaiserslautern
schneider@wiwi.uni-kl.de

Andreas Stenger
Chair of IT-based Logistics, Institute of Information Systems
Goethe University, Frankfurt am Main
stenger@wiwi.uni-frankfurt.de

Dominik Goeke
Chair of Business Information Systems and Operations Research
University of Kaiserslautern
goeke@wiwi.uni-kl.de

**University of Kaiserslautern**
Chair of Business Information Systems and Operations Research (BISOR)
P.O. Box 3049
Erwin-Schrödinger-Straße, Building 42-420
67653 Kaiserslautern, Germany
Email: `bisor@bisor.de`, Web: `http://bisor.de`

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

**Abstract**

Driven by new laws and regulations concerning the emission of greenhouse gases, carriers are starting to use battery electric vehicles (BEVs) for last-mile deliveries. The limited battery capacities of BEVs necessitate visits to recharging stations during delivery tours of industry-typical length, which have to be considered in the route planning in order 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 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.

## 1 Introduction

In recent years, the greenhouse effect has become a hot political topic worldwide and laws and regulations to reduce greenhouse gas pollution have already been passed or are currently under debate. For example, to stop the increasing emissions of light commercial vehicles (<3.5t), EU regulation No 510/2011 imposes 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 (European Parliament and European Council 2011). The white book of the European Commission even envisages a mostly emission-free city logistics until 2030 (European Comission 2011).

Such political decisions and visions have a strong effect on the logistics industry. Many logistics companies have already started to establish "Green Logistics" projects to reduce CO2 emissions. Often, their first step is an increased application of optimization methods to improve route planning, which helps to decrease the traveled distance of their vehicles and hence emissions. However, this generally yields a decline of emissions of only a few percent and the emission level of their trucks and vans remains on a high level.

A more promising alternative is the use of battery electric vehicles (BEVs), which EU regulation No 510/2011 defines to have zero emissions. BEVs failed in earlier years due to exorbitant battery prices and very short driving ranges. However, as BEVs have become one of the major research areas in the automotive sector and more and more BEVs are developed, the magnitude of these problems diminishes. In the small package shipping (SPS) industry, several big companies, like DHL, UPS, DPD and Japan Post, already started using BEVs for last-mile deliveries from depots to customers, in particular in urban areas. This causes new challenges for an efficient route planning due to several specifics of BEVs. For example, the maximum driving range of BEVs is still not sufficient to perform the typical delivery tours of a small package shipper in one run. Since reducing the number of deliveries performed by one vehicle is clearly not a profitable option, visits to recharging stations along the routes are required. The number of available recharging stations is still relatively scarce, which might lead to long detours if the recharging requirements are not integrated into the route planning.

Route planning issues of logistics companies are generally represented as Vehicle Routing Problem (VRP), which seeks to minimize transportation costs for visiting customers, while every customer is visited exactly once and routes start and end at one depot. The original VRP was introduced by Dantzig and Ramser (1959) and 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 (CVRP), where vehicles have a limited freight capacity and the VRP with Time Windows (VRPTW), where customers have to be reached within a specified time interval (Laporte 2009, Nagata et al. 2010). However, to the best of our knowledge, only one routing model that considers recharging stations exists. Erdogan and Miller-Hooks (2012) propose the Green VRP (G-VRP), a routing model for Alternative Fuel Vehicles (AFVs). The G-VRP considers a limited fuel capacity of the vehicles and the possibility to refuel at Alternative Fuel Stations (AFSs). For each refueling as well as for each customer visit, a fixed service time is considered and the maximum duration of a route is restricted.

Logistics providers using BEVs for last-mile deliveries require the incorporation of their most important practical constraints into routing models for electric vehicles. First, vehicle capacity restrictions have to be considered for a significant share of delivery operations. Second, many companies, e.g., in the SPS 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 as recharging times for BEVs 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 SPS companies, 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 incorporates the possibility of recharging at any of the available stations using an appropriate recharging scheme, i.e., recharging times depend on the battery charge of the vehicle on arrival at the station. Moreover, the most important practical requirements of logistics providers using BEVs, namely capacity constraints on vehicles and customer time windows are included. E-VRPTW aims at minimizing the number of employed vehicles and total traveled distance.

As 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 et al. 2012). To solve E-VRPTW, we develop a hybrid metaheuristic, which 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 G-VRP and the Multi-Depot VRP with Inter-Depot Routes (MDVRPI). 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 in order 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 Section 2, a review of related literature is presented. In Section 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 Section 5. Section 6 gives a short summary and conclusion of the paper.

## 2 Literature Review

In this section, we briefly review the literature related to the problem addressed in this paper.

The use of BEVs requires the integration of distance constraints depending on battery charge. Distance constraints in order to include working hour restrictions by assuming the duration to be related to route length by an average speed are quite common in VRPs. Due to the widespread availability of petrol stations and the large cruising range of gasoline powered vehicles, distance constraints, however, have scarcely attracted interest as pure range (fuel) constraints. 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 et al. 1986).

E-VRPTW extends the VRPTW, which is probably the most studied VRP variant in the last two decades. In the VRPTW, service at a customer has to start within a given time interval, which is a highly relevant constraint in real-world routing applications (see, e.g., Bräysy and Gendreau 2005a). Numerous heuristic solution methods has been proposed to solve the VRPTW. Among the best performing are the edge-assembly memetic algorithm of Nagata et al. (2010), the branch-and-price based large neighborhood search of Prescott-Gagnon et al. (2009) and the reactive VNS of Bräysy (2003). For state-of-the-art reviews of heuristic and exact methods for VRPTW, we refer the reader to Gendreau and Tarantilis (2010) and Baldacci et al. (2012).

Another problem that is closely related to E-VRPTW is an extension of the Multi-Depot VRP (MDVRP) described in Crevier et al. (2007). The MDVRP itself is a well-known VRP variant, where vehicles are located at several locally disperse depots and each route has to end at the depot it originated from. The extension by Crevier et al. (2007) is called MDVRP with Inter-Depot Routes (MDVRPI) and is motivated by the deliveries of a grocery in Montreal. The model considers intermediate depots at which vehicles can be replenished with goods during the course of a route. To solve the MDVRPI, Crevier et al. (2007) present a heuristic procedure that combines ideas from adaptive memory programming, described in Rochat and Taillard (1995), TS and integer programming. More precisely, the problem is split into three subproblems: an MDVRP, a VRP and an inter-depot subproblem, for which solutions are determined by means of a TS heuristic and saved in a solution pool. The generated routes are subsequently merged by means of a set-partitioning algorithm, followed by an improvement phase. Although the multi-depot case is described, all proposed benchmark instances consider only one depot at which the vehicle fleet is stationed.

Therefore, Tarantilis et al. (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 a cost-savings heuristics. Second, a VNS algorithm is applied using a TS in the local search phase, instead of a greedy procedure. Third, the solution is further improved by means of a guided local search. In numerical tests performed on available benchmark instances, the heuristic clearly outperforms the solution procedure proposed by Crevier et al. (2007). In addition, they present 54 new benchmark instances with up to 175 customers. Problems similar to VPRIRF arise in the collection of waste, for example, described by Kim et al. (2006). In this context, however, the objective is not only to minimize travel distance but also to balance the workload among the vehicles and to obtain a high route compactness.

Relatively few literature has been published on optimization problems related to alternative fuels. Most articles deal with the question how to place refueling stations in an infrastructure-oriented context, either for refueling vehicles using compressed natural gas (CNG) (Boostani et al. 2010) or electricity (Qiu et al. 2011). The development of an infrastructure consisting of refueling stations, differing in terms of refueling speed and capacity, has been realized to be crucial for the promotion of AFVs. The models usually use a node or flow-based set covering problem to determine the optimal number and location of the refueling stations. To model the fuel demand for short-distance trips in urban areas, customers are usually aggregated to nodes and a node-based formulation is used. Considering long-distance trips within the location decision, the flow between origin-destination pairs is used as a measure for the demand (Wang and Lin 2009, Wang and Wang 2010).

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). Wang and Shen (2007) propose a scheduling problem for electric buses, called Vehicle Scheduling Problem with Route and Fueling Time Constraints. 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 every vehicle has to be recharged after several trips.

Finally, we are aware of three publications that explicitly consider the specific characteristics of alternative fuels and adopt them to VRPs. Gonçalves et al. (2011) consider a VRP with Pickup and Delivery (VRPPD) with a mixed fleet that consists of BEVs and vehicles using internal-combustion engines. The objective is to minimize total costs, which consist of vehicle-related fixed and variable costs. They consider time and capacity constraints and assume a time for recharging the BEVs, which they calculate from the total distance travelled and the range using one battery charge. However, they do not incorporate the actual location of recharging stations into their model. Thus, they basically propose a mixed-fleet VRPPD with an additional distance-dependent time variable.

To the best of our knowledge, Erdogan and Miller-Hooks (2012) are the first to combine a VRP with the possibility of refueling a vehicle at a station along the route. They are mainly motivated by vehicle fleets operating on a wide geographical region and driving with biodiesel, liquid natural gas or CNG. For these fuels only a limited refueling infrastructure exists, but refueling times may be assumed to be fixed. The proposed G-VRP considers a maximum route duration and fuel constraint. Fuel is consumed with a given rate per traveled distance and can be replenished at AFS. In principle, the G-VRP is modeled as an extension to the MDVRPI and Erdogan and Miller-Hooks (2012) propose two heuristics to solve the new problem. The first heuristic is a Modified Clarke and Wright Savings algorithm (MCWS) which creates routes by establishing feasibility through the insertion of AFSs, merging feasible routes according to savings and removing redundant AFSs. The second heuristics is a Density-Based Clustering Algorithm (DBCA) based on a cluster-first and route-second approach. The DBCA forms clusters of customers such that every vertex within a given radius contains at least a predefined number of neighbors. Subsequently, the MCWS algorithm is applied on the identified clusters. For the numerical studies, Erdogan and Miller-Hooks (2012) design two sets of problem instances. The first consists of 40 small-sized instances with 20 customers and the second involves 12 instances with up to 500 customers.

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

Let $V$ be a set of vertices with $V = I \cup F'$, where $I$ 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. Further, let $v_0$ and $v_{n+1}$ denote instances of the same depot, where every route starts at $v_0$ and ends at $v_{n+1}$ and let the indices 0 and $n+1$ indicate that a set contains the respective instance of the depot, e.g., $V_0 = V \cup \{v_0\}$. Then 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 $r \cdot d_{ij}$ of the remaining battery charge of the vehicle traveling the arc, where $r$ denotes the constant charge consumption rate.

At the depot, a set of homogeneous vehicles with a maximal capacity of $C$ are positioned. Each vertex $i \in V_{0,n+1}$ is assigned a positive demand $q_i$, which is set to 0 if $i \notin I$. Moreover, each vertex $i \in V_{0,n+1}$ has a time window $[e_i, l_i]$ and all customers $j \in I$ have an associated service time $s_j$. 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 fuel level of the vehicle when arriving at the respective station.

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 $\tau_j$ specifies the time of arrival, $u_j$ the remaining cargo and $y_j$ the remaining charge level on arrival at vertex $j \in V_{0,n+1}$. The decision variables $x_{ij} \mid i \in V_0, j \in V_{n+1}, i \neq j$ are binary and equal 1 if an arc is traveled and 0 otherwise.

The objective function of E-VRPTW is hierarchical. As commonly done for vehicle routing problems with time window constraints (see, e.g., Bräysy and Gendreau 2005b), our first objective is to minimize the number of vehicles, i.e., a solution with less vehicles is always superior. The second objective is to minimize the total traveled distance.

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

$$\min \sum_{i \in V_0, j \in V_{n+1}, i \neq j} d_{ij} x_{ij} \tag{1}$$

$$\sum_{j \in V_{n+1}, i \neq j} x_{ij} = 1 \quad \forall i \in I \tag{2}$$

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

$$\sum_{i \in V_{n+1}, i \neq j} x_{ji} - \sum_{i \in V_0, i \neq j} x_{ij} = 0 \quad \forall j \in V \tag{4}$$

$$\tau_i + (t_{ij} + s_i) x_{ij} - l_0 (1 - x_{ij}) \leq \tau_j \quad \forall i \in I_0, \forall j \in V_{n+1}, i \neq j \tag{5}$$

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

$$e_j \leq \tau_j \leq l_j \quad \forall j \in V_{0,n+1} \tag{7}$$

$$0 \leq u_j \leq u_i - q_i x_{ij} + C(1 - x_{ij}) \quad \forall i \in V_0, \forall j \in V_{n+1}, i \neq j \tag{8}$$

$$0 \leq u_0 \leq C \tag{9}$$

$$0 \leq y_j \leq y_i - (r \cdot d_{ij}) x_{ij} + Q(1 - x_{ij}) \quad \forall j \in V_{n+1}, \forall i \in I, i \neq j \tag{10}$$

$$0 \leq y_j \leq Q - (r \cdot d_{ij}) x_{ij} \quad \forall j \in V_{n+1}, \forall i \in F'_0, i \neq j \tag{11}$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in V_0, j \in V_{n+1}, i \neq j \tag{12}$$

The objective function is defined in (1). Constraints (2) enforce the connectivity of costumers and Constraints (3) handle the connectivity of visits to recharging station. 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 above, 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) ensure 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 non-negative cargo load upon arrival at any vertex. Finally, Constraints (10) and (11) ensure that the battery charge never falls below zero.

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

As 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., Melechovsky et al.

```
1: $\mathcal{N}_\kappa \leftarrow$ set of VNS neighborhood structures for $\kappa = 1, ..., \kappa_{\max}$
2: Generate initial solution $S$
3: $\kappa \leftarrow 1$
4: $i \leftarrow 0$
5: feasibilityPhase $\leftarrow$ *true*
6: **while** feasibilityPhase $\vee$ ($\neg$ feasibilityPhase $\wedge$ $i < \eta_{dist}$) **do**
7:    $S' \leftarrow$ random point $\in \mathcal{N}_\kappa(S)$)
8:    $S'' \leftarrow$ best solution after $\eta_{tabu}$ iterations of tabu search with $S'$ as initial solution
9:    **if** acceptSA($S''$,$S$) **then**
10:       $S \leftarrow S''$
11:       $\kappa \leftarrow 1$
12:   **else**
13:       $\kappa \leftarrow \kappa + 1$
14:   **end if**
15:   **if** feasibilityPhase **then**
16:       **if** $\neg$ feasible($S$) **then**
17:           **if** $i = \eta_{feas}$ **then**
18:               addVehicle($S$)
19:               $i \leftarrow -1$
20:           **end if**
21:       **else**
22:           feasibilityPhase $\leftarrow$ *false*
23:           $i \leftarrow -1$
24:       **end if**
25:   **end if**
26:   $i \leftarrow i + 1$
27: **end while**
```

Figure 1: Overview of our VNS/TS algorithm for solving E-VRPTW.

2005, Tarantilis et al. 2008). VNS, proposed by Mladenović and Hansen (1997), is an effective metaheuristic performing local search on increasingly larger neighborhoods in order 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 like VRPTW with single or multiple depots (Bräysy 2003, Polacek et al. 2004).

TS is a powerful metaheuristic, which guides local search heuristics to search a solution space economically and effectively (Glover and Laguna 1997). Starting from an initial solution, the best non-tabu move is conducted at each iteration. The diversification of the search is obtained by integrating a memory structure called tabu list, which prevents the search heuristic from cycling. 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 Section 4.1. Infeasible solutions are allowed during the search and evaluated based on a penalizing cost function (see Section 4.2). We first perform a feasibility phase during which the number of vehicles is increased after no feasible solution has been found for a given number of $\eta_{feas}$ iterations. After a feasible solution is found, another $\eta_{dist}$ iterations are performed to improve traveled distance.

The search is guided by a VNS component described in Section 4.3. It uses the current VNS neighborhood $\mathcal{N}_\kappa$ to generate a random perturbation which serves as initial solution for $\eta_{tabu}$ iterations of the TS phase (Section 4.4). The acceptance criterion of the VNS is based on Simulated Annealing (SA).

## 4.1 Preprocessing and Generation of Initial Solution

As commonly done, we apply a preprocessing step to remove infeasible arcs (see, e.g., Psaraftis 1983, Savelsbergh 1985). Arc $(v, w)$ connecting vertices $v$ and $w$ can be removed from the set of possible arcs if one of the following inequalities holds:

$$q_v + q_w \geq C \qquad \forall v, w \in I \tag{13}$$

$$e_v + s_v + t_{vw} \geq l_w \qquad \forall v \in V_0, \forall w \in V_{n+1} \tag{14}$$

$$e_v + s_v + t_{vw} + s_w + t_{wn+1} \geq l_0 \qquad \forall v \in V_0, w \in V \tag{15}$$

$$r(d_{jv} + d_{vw} + d_{wi}) \geq Q \qquad \forall v, j \in V_0, \forall w, i \in V_{n+1} \tag{16}$$

Equation (13) - (15) are well-known preprocessing steps that base 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 labelled infeasible. Numerical studies showed that this preprocessing step is able to strongly reduce the number of feasible arcs on our E-VRPTW test instances.

We construct an initial solution similar to the approach proposed in Cordeau et al. (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. 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 $e_i \leq e_u \leq e_j$. This rule helps to keep time windows but feasibility is only guaranteed concerning capacity and battery capacity for all routes but the last.

## 4.2 Generalized Cost Function

As commonly done in literature, our solution methods allows infeasible solutions during the search process. A solution is evaluated by means of the following generalized cost function:

$$F(S) = L(S) + \alpha P_{cap}(S) + \beta P_{tw}(S) + \gamma P_{batt}(S) + P_{div}(S), \tag{17}$$

where $L(S)$ denote the total traveled distance, $P_{cap}(S)$ the total capacity violation, $P_{tw}(S)$ the time window violation, $P_{batt}(S)$ the battery capacity violation, $P_{div}(S)$ a diversification penalty and $\alpha$, $\beta$ and $\gamma$ are factors weighting the violations. The penalty factors are dynamically updated between a given lower and upper bound. In order to balance between diversification and intensification, they are increased by a factor $\delta$ after the respective constraint has been violated for a certain number of iterations and divided by $\delta$ if the respective constraint was satisfied.

In the following, we describe the efficient calculation of the constraint violations. Let the sequence $v(k) = \langle v_0, v_1, ..., v_n, v_{n+1} \rangle$ contain all ordered vertices of route $k$. Then, the capacity violation of route $k$ can be calculated as

$$P_{cap}(k) = \max\{\sum_{i \in v(k)} d_i - C, 0\},$$

where $v(k)$ refers to the set of customers in route $k$. The total capacity penalty of a solution $S$ is calculated by adding the individual violations of all routes $m$:

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

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 Section 4.4.

To calculate battery capacity violations, we define the following two variables: $\Upsilon^{\rightarrow}_{v_i}$ contains the battery charge that is needed to travel from the last visit to a recharging station or the depot to vertex $v_i$ and $\Upsilon^{\leftarrow}_{v_i}$ is the battery charge that is needed from $v_i$ to the next recharging station or the depot:

$$\Upsilon^{\rightarrow}_{v_i} = \begin{cases} r \cdot d_{v_{i-1}v_i} & \text{if } v_{i-1} \in F'_0 \\ \Upsilon^{\rightarrow}_{v_{i-1}} + r \cdot d_{v_{i-1}v_i} & \text{otherwise} \end{cases} \qquad (i = 1, ..., n+1)$$

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

The battery capacity violation of a route $k$ can then be calculated by adding the individual violations at every visit to a recharging station and on return to the depot:

$$P_{batt}(k) = \sum_{v_i \in v(k) \cap F'_{n+1}} \max\{\Upsilon^{\rightarrow}_{v_i} - Q, 0\}$$

Using the presented variables, changes in battery capacity violation can be calculated in $\mathcal{O}(1)$ for all of the neighborhood operators described in Section 4.4.

To calculate time window violations, we adapt the time window handling approach described in Nagata et al. (2010) and enhanced by Schneider et al. (2012) to E-VRPTW. The approach bases 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 inter-route moves can be calculated in $\mathcal{O}(1)$ for most classical neighborhood structures.

More precisely, by storing forward and backward time window penalty slacks, it is possible to calculate in constant time the time window penalties of a route $k_1 = \langle 0, ..., u, w, ..., 0\rangle$ that is constructed from two partial routes $\langle 0, ..., u\rangle$ and $\langle w, ..., 0\rangle$ or of a route $k_2 = \langle 0, ..., u, v, w, ..., 0\rangle$ that is constructed by inserting a vertex $v$ between two partial routes $\langle 0, ..., u\rangle$ and $\langle w, ..., 0\rangle$. This is not always possible if recharging stations are present as 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 $\langle w, ..., 0\rangle$ contains a recharging station $\vartheta$, i.e., $\langle w, .., \vartheta, \vartheta + 1, .., 0\rangle$, slack variables have to be recalculated by traversing the partial route $\langle w, .., \vartheta + 1\rangle$ for $k_1$ and the partial route $\langle v, .., \vartheta + 1\rangle$ for $k_2$. Note that a recharging station in the first partial route $\langle 0, ..., u\rangle$ or the vertex to insert $v$ 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 as well as the acceptance criterion clearly differs. 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 $\mathcal{N}_{\kappa}$. Our neighborhood structures are all defined by means of the cyclic-exchange operator. In the cyclic-exchange, introduced by Thompson and Orlin (1989), Thompson and Psaraftis (1993), customer sequences of arbitrary length are simultaneously transferred between routes. Our $\kappa$ neighborhood structures, shown in Table 1, are defined according to the following parameters: The number of routes which built the cycle is equal to $\#Rts$. In each route $k$, we randomly select the number of successive vertices that form the translocation chain in the interval $[0, \min\{\Gamma_{\max}, n_k\}]$, where $n_k$ denotes the number of customers and stations contained in $k$. The initial vertex of a chain is randomly chosen in each route. Cyclic-exchange,

| $\kappa$ | $\#Rts$ | $\Gamma_{\max}$ | $\kappa$ | $\#Rts$ | $\Gamma_{\max}$ | $\kappa$ | $\#Rts$ | $\Gamma_{\max}$ |
|---|---|---|---|---|---|---|---|---|
| 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 |

Table 1: The $\kappa$-neighborhood structures used in the VNS defined by the number of involved route $\#Rts$ and the maximum number of translocated vertices $\Gamma_{\max}$.

or a variant that is restricted to two routes, called cross-exchange, are commonly used in the perturbation phase of VNS-algorithms (see, e.g, Polacek et al. 2004, Hemmelmayr et al. 2009).

In the local search phase, we improve the randomly generated solution $S'$ by means of our TS heuristic, detailed in Section 4.4. The search stops after $\eta_{tabu}$ iterations. Note that the perturbation move is added to the general tabu list to prevent its reversal. Subsequently, we compare the best solution found during the local search $S''$ to the initial solution $S$. Instead of accepting only improving solutions, we use an acceptance criterion that is inspired by the metaheuristic SA (Kirkpatrick et al. 1983). This method has been successfully applied in several VNS approaches, for example, in Hemmelmayr et al. (2009) and Stenger et al. (2011).

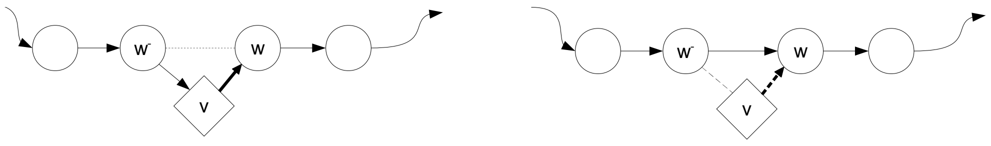
To be more precise, improving solutions are always accepted, while we accept deteriorating solutions according to the Metropolis probability. Let $f(\cdot)$ denote the objective function value, the probability of accepting solution $S''$ is calculated by $e^{\frac{-(f(S'')-f(S))}{\theta}}$. Variable $\theta$ is a system parameter, that is called temperature. At the beginning of the search, the temperature is usually initialized to a high value $\theta_{init}$, so that deteriorating solutions are often accepted, which helps to diversify the search. By continuously decreasing the temperature during the search, an intensification is achieved and, finally, only improving solutions are accepted. In our case, we set $\theta_{init}$ in a way that a solution value $f(S'')$, which is $w_{SA}$ worse than $f(S)$ is accepted with a probability of 50%. After every VNS iteration, the temperature is linearly decreased with a cooling factor $\epsilon$ which is chosen such that the temperature is below 0.0001 during the last 20% of iterations.

## 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 TS is generated by applying the following neighborhood operators on every arc in the list of generator arcs (cp. Toth and Vigo 2003): 2-opt*, relocate, exchange and a new, problem-specific operator called stationInRe. Each move is evaluated and the best non-tabu 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 inter-route 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 station and applied as intra- and inter-route operator. The exchange operator, also introduced in Savelsbergh (1992), swaps the position of two vertices. The operator is applied for inter-route and intra-route 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 either $v$ or $w$ 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 2(a). If the arc is already present, a recharging station is removed as shown in 2(b).



(a) Insertion (b) Removal

Figure 2: Insertion and removal of a recharging station with the stationInRe operator. Generator arcs are shown in bold and removed arcs as dashed lines.

We set every arc $\xi$ that is deleted from the solution by the execution of a move tabu, i.e, we forbid the reinsertion of the arc into the solution for a specified number of iterations called tabu tenure. As station visits have a strong effect on charge levels and also on time windows due to recharging times incurred, we define the tabu attribute $(\xi, k, \mu, \zeta)$. It prohibits the insertion of arc $\xi$ into route $k$ between $\mu$ and $\zeta$, 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 for each arc is randomly drawn from the the interval $[\rho_{\min}, \rho_{\max}]$. The tabu status of a move can be 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 et al. (2001) to E-VRPTW. Using the notation introduced above, we define vertex-based attributes $(u, k, \mu, \zeta)$ to describe that customer/station $u$ is positioned between stations/depot $\mu$ and $\zeta$ in route $k$. In this way, each solution $S$ can be characterized by the attribute set $B_{vertex}(S) = \{(u, k, \mu, \zeta)\}$. 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 $S$, which does not improve the overall best solution, is penalized by:

$$P_{div}(S) = \lambda L(S)\sqrt{nm} \sum_{(u,k,\mu,\zeta)\in B_{vertex}(S)} p_{uk\mu\zeta},$$

where $\lambda$ is a parameter to control the amount of diversification and the scaling factor $L(S)\sqrt{nm}$ establishes a relation between the diversification penalty and the traveled distance and the investigated problem size with $n$ customers and $m$ vehicles. The TS procedure is stopped after $\eta_{tabu}$ iterations.

## 5 Numerical Experiments

In this section, we present the extensive numerical testing conducted to evaluate the performance of our hybrid VNS/TS method. The first study evaluates the performance of our VNS/TS on E-VRPTW instances. To be able to asses the solution quality, we use newly designed small instances which 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 heuristics, namely the VNS, TS and SA, on a set of medium-sized E-VRPTW instances, which we designed 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 and G-VRP.

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

### 5.1 Experimental Environment & Parameter Settings

All tests are performed on a desktop computer equipped with an Intel Core i5 processor with 2.67 GHz and 4 GB RAM, operating Windows 7 Professional. The VNS/TS is implemented as single-thread code in java. The parameters we used to generate the final results are provided in Table 2. The presented values are the result of intensive studies we conducted to fine-tune our algorithm. In the following, we briefly discuss only those parameters that have a strong effect on the performance of our algorithm.

| Phases | | Penalties | | Tabu list | | Conti. Div. | | VNS | |
|---|---|---|---|---|---|---|---|---|---|
| $\eta_{feas}$ | 500 | $\alpha_0, \beta_0, \gamma_0$ | 10 | $\rho_{\min}$ | 15 | $\lambda$ | 1.0 | $\eta_{tabu}$ | 100 |
| $\eta_{dist}$ | 200 | $\alpha_{\min}, \beta_{\min}, \gamma_{\min}$ | 0.5 | $\rho_{\max}$ | 30 | | | $w_{SA}$ | 8% |
| | | $\alpha_{\max}, \beta_{\max}, \gamma_{\max}$ | 5000 | | | | | | |
| | | $\delta$ | 1.2 | | | | | | |
| | | $\eta_{penalty}$ | 2 | | | | | | |

Table 2: Overview of the parameter settings chosen for the numerical studies.

During our testing, we observed that the values chosen for the initial penalty factors $\alpha_0, \beta_0, \gamma_0$ are crucial for the solution quality of our VNS/TS. In case of high values, the search often got stuck in low-quality local minima and required several iterations before continuing an effective search of solution space. By contrast, setting the initial value too low favors the acceptance of highly infeasible solutions, which also has a negative influence on the performance. We obtained best results with a value of 10, which seems a good compromise between diversification and intensification at the beginning of the search. During the search, the penalty factors are updated by multiplying or dividing by factor 1.2, while limiting the values to the interval $[0.5, 5000]$.

Concerning the feasibility phase, the tests showed that if the VNS/TS is not able to find a feasible solution with the given number of vehicles in 500 VNS iterations, it is very unlikely that a feasible solution with this vehicle number is found in later iterations. The entire algorithm terminates after 200 additional distance minimization iterations as this resulted in a good trade-off between computing time and solution quality. Furthermore, the length of the tabu list clearly affected the performance of our algorithm. However, we were not able to find a unique value that performed well on all instances of the different benchmark sets that we solved. Instead, we achieved the overall best results by randomly selecting the length from the interval $[15, 30]$ in each iteration.

## 5.2 Experiments on E-VRPTW Instances

As 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, which we describe in Section 5.2.1. Section 5.2.2 presents the results of our testing 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 customer per instance. All instances are created based on the benchmark instances for the VRPTW proposed by Solomon (1987). These instances are divided into 3 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 such that the feasibility of the instance is guaranteed, i.e., that 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) 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 disperse 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 to 1.0. The inverse refueling rate is set to a value so that a complete refueling requires three times the average customer service time of the respective instance.

Since the limited battery capacity and the need for recharging along the routes lead to longer route durations, the customer time windows given in the original VRPTW instances have to be altered, as instances with tight time windows would otherwise become infeasible. To this end, we determine new time windows following the procedure 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 for download.

To generate the set of small instances, we first generate 168 instances of three sizes ($5, 10, 15$ customers) by randomly drawing the respective number of customer from each of the large instances. The created instances are solved with our VNS/TS heuristic and the solutions are inspected. For each problem 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.

### 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 ILOG CPLEX 12.2, using the E-VRPTW formulation presented in Section 3. Table

3 provides an overview of the results. For both, CPLEX and our heuristic, we provide the number of vehicles required in column $m$ and the computing time in seconds in column $t(s)$. For the solutions obtained with CPLEX, the objective function value given in column $L_{best}(S)$ corresponds to the optimal solution, or the best upper bound found within 7200 seconds. For VNS/TS, this column provides the best solution found in 10 runs and column $\Delta$ denotes the gap to the solution found by CPLEX.

| | **CPLEX** | | | **VNS/TS** | | | |
|---|---|---|---|---|---|---|---|
| | $m$ | $L_{best}(S)$ | $t(sec)$ | $m$ | $L_{best}(S)$ | $\Delta_{best}(\%)$ | $t(sec)$ |
| C101C5 | 2 | 257.75 | 81 | 2 | 257.75 | 0.00 | 0.21 |
| C103C5 | 1 | 176.05 | 5 | 1 | 176.05 | 0.00 | 0.12 |
| C206C5 | 1 | 242.56 | 518 | 1 | 242.55 | 0.00 | 0.14 |
| C208C5 | 1 | 158.48 | 15 | 1 | 158.48 | 0.00 | 0.11 |
| R104C5 | 2 | 136.69 | 1 | 2 | 136.69 | 0.00 | 0.13 |
| R105C5 | 2 | 156.08 | 3 | 2 | 156.08 | 0.00 | 0.11 |
| R202C5 | 1 | 128.78 | 1 | 1 | 128.78 | 0.00 | 0.11 |
| R203C5 | 1 | 179.06 | 5 | 1 | 179.06 | 0.00 | 0.15 |
| RC105C5 | 2 | 241.3 | 764 | 2 | 241.3 | 0.00 | 0.14 |
| RC108C5 | 1 | 253.93 | 311 | 2 | 253.93 | 0.00 | 0.17 |
| RC204C5 | 1 | 176.39 | 54 | 1 | 176.39 | 0.00 | 0.15 |
| RC208C5 | 1 | 167.98 | 21 | 1 | 167.98 | 0.00 | 0.13 |
| C101C10 | 3 | 393.76 | 171 | 3 | 393.76 | 0.00 | 0.77 |
| C104C10 | 2 | 273.93 | 360 | 2 | 273.93 | 0.00 | 0.95 |
| C202C10 | 1 | 304.06 | 300 | 1 | 304.06 | 0.00 | 0.71 |
| C205C10 | 2 | 228.28 | 4 | 2 | 228.28 | 0.00 | 0.49 |
| R102C10 | 3 | 249.19 | 389 | 3 | 249.19 | 0.00 | 0.65 |
| R103C10 | 2 | 207.05 | 119 | 2 | 207.05 | 0.00 | 0.72 |
| R201C10 | 1 | 241.51 | 177 | 1 | 241.51 | 0.00 | 0.78 |
| R203C10 | 1 | 218.21 | 573 | 1 | 218.21 | 0.00 | 0.71 |
| RC102C10 | 4 | 423.51 | 810 | 4 | 423.51 | 0.00 | 0.69 |
| RC108C10 | 3 | 345.93 | 39 | 3 | 345.93 | 0.00 | 0.9 |
| RC201C10 | 1 | *412.86* | 7200 | 1 | 412.86 | 0.00 | 0.9 |
| RC205C10 | 2 | 325.98 | 399 | 2 | 325.98 | 0.00 | 0.81 |
| C103C15 | 3 | *384.29* | 7200 | 3 | 384.29 | 0.00 | 15.37 |
| C106C15 | 3 | 275.13 | 17 | 3 | 275.13 | 0.00 | 14.94 |
| C202C15 | 2 | *383.62* | 7200 | 2 | 383.61 | 0.00 | 13.41 |
| C208C15 | 2 | 300.55 | 5060 | 2 | 300.55 | 0.00 | 11.08 |
| R102C15 | 5 | *413.93* | 7200 | 5 | 413.93 | 0.00 | 19.55 |
| R105C15 | 4 | *336.15* | 7200 | 4 | 336.15 | 0.00 | 13.35 |
| R202C15 | 2 | 358 | 7200 | 2 | 358 | 0.00 | 13.17 |
| R209C15 | 1 | *313.24* | 7200 | 1 | 313.24 | 0.00 | 13.73 |
| RC103C15 | 4 | *397.67* | 7200 | 4 | 397.67 | 0.00 | 14.62 |
| RC108C15 | 3 | *370.25* | 7200 | 3 | 370.25 | 0.00 | 12.92 |
| RC202C15 | 2 | *394.39* | 7200 | 2 | 394.39 | 0.00 | 12.74 |
| RC204C15 | 1 | *407.45* | 7200 | 1 | 384.86 | -5.87 | 15.57 |
| Average | | | 2483.25 | | | -0.16 | 5.03 |

Table 3: Comparison of results obtained with CPLEX and VNS/TS on the small-sized instances. $L_{best}(S)$ denotes the best found solution with the minimal number of vehicles $m$. $t(s)$ denotes the total runtime in seconds. The maximum duration for CPLEX was set to 2 hours, so optimality is not guaranteed for CPLEX results which used the full time.

The results clearly show the ability of our VNS/TS heuristic to solve small E-VRPTW instances to optimality in only a few seconds. Independent of the instance structure or size, we always obtain the optimal solution, if CPLEX found an optimum within 7200s. For most of the 15-customer instances and one 10-customer instance, CPLEX was not able to provide the optimal solution. On those instances, we either found a solution equal to the upper bound provided by CPLEX or a better solution in one case.

### 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 instances to the solutions found with 1) a VNS/TS heuristic that accepts only improving solution after the local search phase instead of using an SA-based criterion (VNS/TS w/o SA) and 2) a pure TS heuristic.

An overview of the results is given in Table 4. For each heuristic, we provide the best solution found in 10 runs ($L_{best}(S)$) and the number $m$ of required vehicles. Furthermore, we determine gaps to the best solution found during the overall testing ($BKS$) for both the objective function value ($\Delta L$) and the number of vehicles ($\Delta m$). Finally, at the bottom of the table, the average computing time in minutes is reported in row $t(min)$.

| | BKS | | VNS/TS | | | | VNS/TS w/o SA | | | | TS | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Instance | $m$ | $L_{best}(S)$ | $m$ | $L_{best}(S)$ | $\Delta_m$ | $\Delta_L(\%)$ | $m$ | $L_{best}(S)$ | $\Delta_m$ | $\Delta_L(\%)$ | $m$ | $L_{best}(S)$ | $\Delta_m$ | $\Delta_L(\%)$ |
| c101 | 12 | 1053.83 | 12 | 1053.83 | 0 | 0.00 | 12 | 1053.83 | 0 | 0.00 | 12 | 1053.83 | 0 | 0.00 |
| c102 | 11 | 1056.47 | 11 | 1057.16 | 0 | 0.07 | 11 | 1056.47 | 0 | 0.00 | 11 | 1069.35 | 0 | 1.22 |
| c103 | 10 | 1041.55 | 10 | 1041.55 | 0 | 0.00 | 11 | 1002.03 | 1 | -3.79 | 10 | 1134.36 | 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 | 1075.37 | 11 | 1075.37 | 0 | 0.00 | 11 | 1075.37 | 0 | 0.00 | 11 | 1079.69 | 0 | 0.40 |
| c106 | 11 | 1057.87 | 11 | 1057.87 | 0 | 0.00 | 11 | 1057.87 | 0 | 0.00 | 11 | 1057.87 | 0 | 0.00 |
| c107 | 11 | 1031.56 | 11 | 1031.56 | 0 | 0.00 | 11 | 1031.56 | 0 | 0.00 | 11 | 1033.08 | 0 | 0.15 |
| c108 | 10 | 1100.32 | 10 | 1100.32 | 0 | 0.00 | 11 | 1015.73 | 1 | -7.69 | 11 | 1015.73 | 1 | -7.69 |
| c109 | 10 | 1036.64 | 10 | 1051.84 | 0 | 1.47 | 10 | 1036.64 | 0 | 0.00 | 10 | 1051.36 | 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 | 1670.8 | 18 | 1672.55 | 0 | 0.10 | 18 | 1673.12 | 0 | 0.14 | 18 | 1670.8 | 0 | 0.00 |
| r102 | 16 | 1495.31 | 16 | 1535.81 | 0 | 2.71 | 16 | 1522.84 | 0 | 1.84 | 16 | 1495.31 | 0 | 0.00 |
| r103 | 13 | 1299.17 | 13 | 1299.64 | 0 | 0.04 | 13 | 1299.17 | 0 | 0.00 | 13 | 1348.25 | 0 | 3.78 |
| r104 | 11 | 1088.43 | 11 | 1088.43 | 0 | 0.00 | 11 | 1143.69 | 0 | 5.08 | 11 | 1097.09 | 0 | 0.80 |
| r105 | 14 | 1461.25 | 14 | 1473.59 | 0 | 0.84 | 15 | 1401.24 | 1 | -4.11 | 14 | 1514.36 | 0 | 3.63 |
| r106 | 13 | 1344.66 | 13 | 1344.66 | 0 | 0.00 | 13 | 1395.18 | 0 | 3.76 | 13 | 1369.55 | 0 | 1.85 |
| r107 | 12 | 1154.52 | 12 | 1154.52 | 0 | 0.00 | 12 | 1158.13 | 0 | 0.31 | 12 | 1162.9 | 0 | 0.73 |
| r108 | 11 | 1050.04 | 11 | 1065.89 | 0 | 1.51 | 11 | 1061.91 | 0 | 1.13 | 11 | 1056.84 | 0 | 0.65 |
| r109 | 12 | 1294.05 | 12 | 1294.05 | 0 | 0.00 | 12 | 1341.01 | 0 | 3.63 | 12 | 1308.62 | 0 | 1.13 |
| r110 | 11 | 1126.74 | 11 | 1143.52 | 0 | 1.49 | 11 | 1141.9 | 0 | 1.35 | 11 | 1126.74 | 0 | 0.00 |
| r111 | 12 | 1106.19 | 12 | 1124.06 | 0 | 1.62 | 12 | 1107.52 | 0 | 0.12 | 12 | 1123.96 | 0 | 1.61 |
| r112 | 11 | 1026.52 | 11 | 1026.52 | 0 | 0.00 | 11 | 1033.97 | 0 | 0.73 | 11 | 1047.92 | 0 | 2.08 |
| r201 | 3 | 1264.82 | 3 | 1264.82 | 0 | 0.00 | 3 | 1264.82 | 0 | 0.00 | 3 | 1266.26 | 0 | 0.11 |
| r202 | 3 | 1052.32 | 3 | 1052.32 | 0 | 0.00 | 3 | 1053.11 | 0 | 0.08 | 3 | 1052.65 | 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 | 1000.96 | 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 | 1731.07 | 16 | 1731.07 | 0 | 0.00 | 16 | 1766.44 | 0 | 2.04 | 16 | 1753.35 | 0 | 1.29 |
| rc102 | 15 | 1554.61 | 15 | 1554.61 | 0 | 0.00 | 15 | 1556.08 | 0 | 0.09 | 15 | 1559.95 | 0 | 0.34 |
| rc103 | 13 | 1351.15 | 13 | 1353.55 | 0 | 0.18 | 13 | 1351.15 | 0 | 0.00 | 13 | 1355.36 | 0 | 0.31 |
| rc104 | 11 | 1238.56 | 11 | 1249.23 | 0 | 0.86 | 11 | 1267.55 | 0 | 2.34 | 11 | 1280.82 | 0 | 3.41 |
| rc105 | 14 | 1475.31 | 14 | 1483.38 | 0 | 0.55 | 14 | 1475.31 | 0 | 0.00 | 14 | 1479.56 | 0 | 0.29 |
| rc106 | 13 | 1437.96 | 13 | 1440.19 | 0 | 0.15 | 13 | 1469.99 | 0 | 2.23 | 13 | 1437.96 | 0 | 0.00 |
| rc107 | 12 | 1279.08 | 12 | 1275.89 | 0 | 0.00 | 12 | 1280.44 | 0 | 0.36 | 12 | 1284.47 | 0 | 0.67 |
| rc108 | 11 | 1209.61 | 11 | 1238.81 | 0 | 2.41 | 11 | 1227.88 | 0 | 1.51 | 11 | 1209.61 | 0 | 0.00 |
| rc201 | 4 | 1444.94 | 4 | 1447.2 | 0 | 0.16 | 4 | 1444.94 | 0 | 0.00 | 4 | 1446.03 | 0 | 0.08 |
| rc202 | 3 | 1418.79 | 3 | 1412.91 | 0 | 0.00 | 3 | 1418.79 | 0 | 0.42 | 3 | 1425.17 | 0 | 0.87 |
| rc203 | 3 | 1073.98 | 3 | 1078.28 | 0 | 0.40 | 3 | 1077.16 | 0 | 0.30 | 3 | 1084.66 | 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 | 1330.53 | 3 | 1321.75 | 0 | 0.00 | 3 | 1353.54 | 0 | 2.41 | 3 | 1360.39 | 0 | 2.92 |
| rc206 | 3 | 1190.75 | 3 | 1191.13 | 0 | 0.03 | 3 | 1204.93 | 0 | 1.19 | 3 | 1207.77 | 0 | 1.43 |
| rc207 | 3 | 1004.38 | 3 | 995.52 | 0 | 0.00 | 3 | 1015.6 | 0 | 2.02 | 3 | 1010.66 | 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 |
| Average | | | | | 0 | 0.35 | | | 0.05 | 0.46 | | | 0.02 | 0.75 |
| $t(min)$ | | | | | 15.34 | | | | 16.22 | | | | 16.01 | |

Table 4: Comparison of the effect of different heuristic components: VNS/TS denotes the standard setting of a hybrid VNS with 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. Gaps are calculated to the best known solution (BKS).

The results show that the VNS/TS heuristic performs best with an average gap of 0.35% to the BKS. A comparison to the results obtained with VNS/TS w/o SA allows to quantify the impact of the SA-based acceptance criterion. Using SA instead of simply accepting improving solutions yields a reduction of the gap of 0.1% on average. Comparing the results to those of the pure TS, we can see that the hybridization of VNS and TS reduces the gap to the best solution by more than half. Overall, the results show the positive effect of the hybridization of VNS, TS and SA. The solution quality is improved by every component incorporated into our heuristic, while computing times remain stable on a moderate level.

## 5.3 Performance on Benchmark Instances of Related Problems

The E-VRPTW is closely related to the MDVRPI and the G-VRP. For both problems, sets of benchmark instances exist. To demonstrate the performance of our VNS/TS heuristic on large problem sets, 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.

### 5.3.1 MDVRPI

For the MDVRPI (respectively the VRPIF), two benchmark sets with a total of 76 instances are available from the literature. The first set of benchmark instances was proposed by Crevier et al. (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 et al. (2008) and involves 54 instances. The set consists of 18 depot-customer combinations, which were created following the design described in Crevier et al. (2007) and compromise 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 all instances, travel time is assumed to be equal to travel distance and distances between vertices are computed as Euclidean distances from the given coordinates.

In Table 5, we compare the results obtained with our VNS/TS on the instance set of Crevier et al. (2007) to the solutions of the heuristic proposed in Tarantilis et al. (2008), denoted as HGL, and in Crevier et al. (2007), denoted as CCL. For CCL and our VNS/TS, we provide the best solution found in 10 runs ($L_{best}(S)$) and the computing time in minutes ($t(min)$). By contrast, the value given in column $L_{best}(S)$ for HGL corresponds to the best solution ever found with the final parameter setting. We further provide the gap of the best ($\Delta_{best}$) and average solution ($\Delta_{avg}$) to the best known solution ($BKS$). Additionally, the last column ($\overline{\text{VNS/TS}}$) shows the best solutions that we found with our VNS/TS heuristic during the overall testing as well as the percentage improvement 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, while 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 et al. (2008) solved only the first subset of instances with their HGL approach. Compared to their results, we are on average 0.48% worse, however, a direct comparison is not fair, since the HGL results correspond to the best solution they ever found during their testing.

Table 6 compares the results of our VNS/TS heuristic with those of HGL on the instance set of Tarantilis et al. (2008). On those instances, our VNS/TS heuristic shows a really strong performance. Our results are on average 0.05% better than those of HGL. 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 HGL. 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 G-VRP

The benchmark instances for the G-VRP were proposed in Erdogan and Miller-Hooks (2012) and consist of four sets, each involving ten instances with 20 customers each. The instances differ in the customer distribution (random or clustered) and the number of available AFS (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 small 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

| | | **CCL** | | | | **HGL** | | | | **VNS/TS** | | | | $\overline{\textbf{VNS/TS}}$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Inst. | BKS | $L_{best}(S)$ | $\Delta_{best}$ (%) | $\Delta_{avg}$ (%) | $t(min)$ | $L_{best}(S)*$ | $\Delta_{best}*$ (%) | $\Delta_{avg}$ (%) | $t(min)$ | $L_{best}(S)$ | $\Delta_{best}$ (%) | $\Delta_{avg}$ (%) | $t(min)$ | $L_{best}(S)$ | $\Delta_{best}$ (%) |
| a1 | **1179.79** | 1203.39 | 2.00 | 2.67 | 4.58 | **1179.79** | 0.00 | 0.84 | 3.4 | **1179.79** | 0.00 | 1.51 | 1.82 | **1179.79** | 0.00 |
| b1 | **1217.07** | 1217.07 | 0.00 | 1.28 | 9.17 | **1217.07** | 0.00 | 0.66 | 7.8 | **1217.07** | 0.00 | 0.80 | 7.14 | **1217.07** | 0.00 |
| c1 | **1883.05** | 1888.22 | 0.27 | 0.53 | 36.22 | **1883.05** | 0.00 | 0.84 | 34.2 | 1897.3 | 0.76 | 2.24 | 33.93 | 1897.3 | 0.76 |
| d1 | **1059.43** | 1059.43 | 0.00 | 1.59 | 8.55 | **1059.43** | 0.00 | 0.46 | 5.9 | 1060.1 | 0.06 | 0.34 | 1.82 | **1059.43** | 0.00 |
| e1 | **1309.12** | **1309.12** | 0.00 | 0.19 | 13.52 | **1309.12** | 0.00 | 0.00 | 8.7 | **1309.12** | 0.00 | 2.66 | 7.29 | **1309.12** | 0.00 |
| f1 | **1572.17** | 1592.25 | 1.28 | 1.87 | 41.41 | **1572.17** | 0.00 | 0.87 | 38.8 | 1584.06 | 0.76 | 3.03 | 34.61 | 1575.57 | 0.22 |
| g1 | **1181.13** | 1190.93 | 0.83 | 1.77 | 55.22 | **1181.13** | 0.00 | 0.77 | 5.8 | 1181.99 | 0.07 | 0.81 | 4.21 | **1181.13** | 0.00 |
| h1 | **1547.25** | 1566.75 | 1.26 | 3.31 | 32.07 | **1547.24** | 0.00 | 1.96 | 11.1 | 1566.19 | 1.22 | 2.27 | 18.03 | 1562.56 | 0.99 |
| i1 | **1925.99** | 1945.73 | 1.02 | 2.60 | 51.01 | **1925.99** | 0.00 | 1.57 | 42.5 | 1953.39 | 1.42 | 4.07 | 45.62 | 1933.05 | 0.37 |
| j1 | 1117.20 | 1144.41 | 2.44 | 3.99 | 58.90 | 1117.20 | 0.00 | 1.04 | 5.5 | **1115.78** | -0.13 | 0.31 | 4.24 | **1115.78** | -0.13 |
| k1 | **1580.39** | 1586.92 | 0.41 | 2.41 | 64.61 | **1580.39** | 0.00 | 0.72 | 12.1 | 1586.64 | 0.40 | 1.35 | 18.11 | 1580.92 | 0.03 |
| l1 | **1880.60** | 1897.74 | 0.91 | 1.94 | 104.27 | **1880.60** | 0.00 | 1.27 | 51.4 | 1902.72 | 1.18 | 2.78 | 46.14 | 1894.05 | 0.72 |
| | | Avg. | 0.87 | 2.01 | 39.96 | | 0.00 | 0.92 | 18.92 | | 0.48 | 1.85 | 18.58 | | |
| a2 | **997.94** | 1000.24 | 0.23 | 0.72 | 6.4 | | | | | **997.94** | 0.00 | 0.47 | 1.8 | **997.94** | 0.00 |
| b2 | 1307.28 | 1307.28 | 0.00 | 1.98 | 14.7 | | | | | 1301.21 | -0.46 | 1.34 | 7.35 | **1291.19** | -1.23 |
| c2 | 1747.61 | 1751.45 | 0.22 | 2.57 | 61.7 | | | | | 1732.19 | -0.88 | 0.76 | 18.05 | **1719.47** | -1.61 |
| d2 | 1871.42 | 1877.03 | 0.30 | 1.43 | 40.5 | | | | | 1892.62 | 1.13 | 1.97 | 35.1 | **1866.97** | -0.24 |
| e2 | 1942.85 | 1974.13 | 1.61 | 2.72 | 73.8 | | | | | 1940.52 | -0.12 | 2.61 | 59.12 | **1928.06** | -0.76 |
| f2 | 2284.35 | 2298.51 | 0.62 | 1.22 | 162.2 | | | | | 2292.4 | 0.35 | 1.79 | 89.86 | **2275.28** | -0.40 |
| g2 | 1162.58 | 1162.58 | 0.00 | 2.01 | 29.5 | | | | | 1158.21 | -0.38 | -0.09 | 4.14 | **1152.92** | -0.83 |
| h2 | 1587.37 | 1593.40 | 0.38 | 1.54 | 160.8 | | | | | 1597.41 | 0.63 | 1.46 | 18.35 | **1580.55** | -0.43 |
| i2 | 1972.00 | 1978.70 | 0.34 | 1.33 | 322.4 | | | | | 1934.09 | -1.92 | -0.10 | 47.58 | **1925.52** | -2.36 |
| j2 | 2294.06 | 2303.01 | 0.39 | 1.36 | 256.9 | | | | | 2293.4 | -0.03 | 1.58 | 91.3 | **2276.52** | -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 | | |

* Note that this value corresponds to the best solution ever obtained with the final parameter setting

Table 5: Comparison of the solutions obtained on the MDVRPI instances, proposed by Crevier et al. (2007), to those of HGL and CCL. BKS denotes the previously best known solution. Gaps are calculated in dependence of BKS. Additionally, we provide the best solutions in $\overline{\text{VNS/TS}}$ that we ever obtained on the instances during our testing activities. $t(min)$ denotes the average runtime for each run.

one halt at an AFS 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 ILOG CPLEX. The CPLEX solution is, however, not the optimal solution to the instance. In their mathematical formulation, they fixed the number of vehicles required to the value obtained with the best heuristic in order 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. In Table 7, we report the best upper bound found in at most 3 hours of computing time. 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 solution found ($L_{best}(S)$) as well as the average computing time in minutes ($t(min)$). We compare our results to those obtained with the MCWS and the DBCA heuristic. Unfortunately, no computing times are available for those heuristics. In the table, we further report the gap of the best solution to the solution found by CPLEX ($\Delta$). Column $n$ provides the number of feasible customers, and $m$ the number of vehicles required.

Our VNS/TS heuristic clearly outperforms both heuristic methods proposed by Erdogan and Miller-Hooks (2012), which both achieve an average gap to best solution of about 8%. On all instances, we obtain the best solution found by CPLEX or even a solution that improves on the upper bound, resulting in an average gap of -0.09%. It is also worth mentioning that we are able to 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 8, we compare the results obtained with our heuristic to those of MCWS and DBCA. The value given in column $\Delta$ denotes the gap to the best solution found, reported

| | | HGL | | | | | VNS/TS | | | | | $\overline{\text{VNS/TS}}$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Instance | BKS | $L_{best}(S)*$ | $\Delta_{best}$* (%) | $L_{avg}(S)$ | $\Delta_{avg}$ (%) | $t(min)$ | $L_{best}(S)$ | $\Delta_{best}$ (%) | $L_{avg}(S)$ | $\Delta_{avg}$ (%) | $t(min)$ | $L_{best}(S)$ | $\Delta_{best}$ (%) |
| 50c3d2v | **2209.83** | **2209.83** | 0.00 | 2260.02 | 2.27 | 2.85 | **2209.83** | 0.00 | 2212.08 | 0.10 | 1.82 | **2209.83** | 0.00 |
| 50c3d4v | **2368.33** | **2368.33** | 0.00 | 2419.86 | 2.18 | 2.23 | **2368.33** | 0.00 | 2389.5 | 0.89 | 1.84 | **2368.33** | 0.00 |
| 50c3d6v | 3000.88 | 3000.88 | 0.00 | 3064.71 | 2.13 | 2.74 | **2999.29** | -0.05 | 3023.46 | 0.75 | 1.88 | **2999.29** | -0.05 |
| 50c5d2v | **2608.25** | **2608.25** | 0.00 | 2683 | 2.87 | 1.54 | **2608.25** | 0.00 | 2634.63 | 1.01 | 2 | **2608.25** | 0.00 |
| 50c5d4v | **3086.58** | **3086.58** | 0.00 | 3124.59 | 1.23 | 2.07 | **3086.58** | 0.00 | 3086.58 | 0.00 | 1.98 | **3086.58** | 0.00 |
| 50c5d6v | 3552 | 3552 | 0.00 | 3583.9 | 0.90 | 3.04 | 3552 | 0.00 | 3562.07 | 0.28 | 2.02 | **3548.88** | -0.09 |
| 50c7d2v | **3353.08** | **3353.08** | 0.00 | 3432.68 | 2.37 | 3.16 | 3353.83 | 0.02 | 3457.14 | 3.10 | 2.38 | **3353.08** | 0.00 |
| 50c7d4v | 3381.57 | 3381.57 | 0.00 | 3470.6 | 2.63 | 3.36 | **3380.27** | -0.04 | 3399.73 | 0.54 | 2.1 | **3380.27** | -0.04 |
| 50c7d6v | 4097.8 | 4097.8 | 0.00 | 4108.1 | 0.25 | 3.42 | **4074.44** | -0.57 | 4113.05 | 0.37 | 2.07 | **4074.44** | -0.57 |
| 75c3d2v | **2678.8** | **2678.8** | 0.00 | 2694.04 | 0.57 | 4.5 | 2692.76 | 0.52 | 2723.84 | 1.68 | 4.67 | **2678.8** | 0.00 |
| 75c3d4v | **2746.74** | **2746.74** | 0.00 | 2800.41 | 1.95 | 3.38 | 2746.74 | 0.00 | 2751.43 | 0.17 | 4.33 | **2746.74** | 0.00 |
| 75c3d6v | 3454.71 | 3454.71 | 0.00 | 3499.54 | 1.30 | 4.89 | 3448.64 | -0.18 | 3468.77 | 0.41 | 4.34 | **3404.34** | -1.46 |
| 75c5d2v | **3373.69** | **3373.69** | 0.00 | 3474.37 | 2.98 | 3.29 | 3386.64 | 0.38 | 3452.66 | 2.34 | 5.09 | **3373.69** | 0.00 |
| 75c5d4v | 3568.35 | 3568.35 | 0.00 | 3655.04 | 2.43 | 3.54 | 3569.82 | 0.04 | 3590.88 | 0.63 | 4.42 | **3553.46** | -0.42 |
| 75c5d6v | 4198.61 | 4198.61 | 0.00 | 4268.19 | 1.66 | 4.18 | 4215.3 | 0.40 | 4284.36 | 2.04 | 4.55 | **4193.86** | -0.11 |
| 75c7d2v | **3569.02** | **3569.02** | 0.00 | 3655.05 | 2.41 | 5.38 | 3581.32 | 0.34 | 3627.34 | 1.63 | 5.06 | **3569.02** | 0.00 |
| 75c7d4v | 3830.43 | 3830.43 | 0.00 | 3911.89 | 2.13 | 5.51 | 3830.43 | 0.00 | 3895.67 | 1.70 | 4.61 | **3825.37** | -0.13 |
| 75c7d6v | **4239.76** | **4239.76** | 0.00 | 4325.33 | 2.02 | 4.29 | 4244.35 | 0.11 | 4271.7 | 0.75 | 4.8 | 4242.08 | 0.05 |
| 100c3d3v | **3123.51** | **3123.51** | 0.00 | 3157.96 | 1.10 | 7.01 | 3127.65 | 0.13 | 3196.39 | 2.33 | 7.94 | 3126.55 | 0.10 |
| 100c3d5v | 3552.5 | 3552.5 | 0.00 | 3636.56 | 2.37 | 7.31 | **3548.75** | -0.11 | 3558.91 | 0.18 | 7.62 | **3548.44** | -0.11 |
| 100c3d7v | 4239.83 | 4239.83 | 0.00 | 4274.86 | 0.83 | 6.62 | 4268.34 | 0.67 | 4339.03 | 2.34 | 7.92 | **4239.5** | -0.01 |
| 100c5d3v | **4053.95** | **4053.95** | 0.00 | 4096.98 | 1.06 | 7.88 | **4053.95** | 0.00 | 4118.03 | 1.58 | 8.49 | **4053.95** | 0.00 |
| 100c5d5v | **4413.17** | **4413.17** | 0.00 | 4531.9 | 2.69 | 7.2 | 4424.81 | 0.26 | 4656.75 | 5.52 | 7.7 | 4415.48 | 0.05 |
| 100c5d7v | 5148.98 | 5148.98 | 0.00 | 5178.02 | 0.56 | 7.72 | **5142.52** | -0.13 | 5156.9 | 0.15 | 7.93 | **5142.52** | -0.13 |
| 100c7d3v | **4216.47** | **4216.47** | 0.00 | 4242.24 | 0.61 | 8.53 | 4242.38 | 0.61 | 4284.85 | 1.62 | 8.87 | **4216.47** | 0.00 |
| 100c7d5v | 4462.51 | 4462.51 | 0.00 | 4523.01 | 1.36 | 8.79 | 4448.15 | -0.32 | 4492.44 | 0.67 | 8 | **4439.72** | -0.51 |
| 100c7d7v | 4897.47 | 4897.47 | 0.00 | 4973.37 | 1.55 | 8.35 | 4916.62 | 0.39 | 5084.82 | 3.83 | 8.1 | **4869.66** | -0.57 |
| 125c4d3v | 3920.05 | 3920.05 | 0.00 | 3966.16 | 1.18 | 8.73 | 3966.61 | 1.19 | 4061.97 | 3.62 | 13.23 | **3916.02** | -0.10 |
| 125c4d5v | 4315.68 | 4315.68 | 0.00 | 4371.7 | 1.30 | 9 | **4308.44** | -0.17 | 4327.81 | 0.28 | 12.33 | **4308.44** | -0.17 |
| 125c4d7v | 4763.49 | 4763.49 | 0.00 | 4833.91 | 1.48 | 8.4 | 4694.32 | -1.45 | 4790.8 | 0.57 | 12.54 | **4668.77** | -1.99 |
| 125c6d3v | **4064.2** | **4064.2** | 0.00 | 4095.72 | 0.78 | 9.19 | 4117.41 | 1.31 | 4202.41 | 3.40 | 13.56 | 4076.04 | 0.29 |
| 125c6d5v | 4826.71 | 4826.71 | 0.00 | 4956.95 | 2.70 | 8.33 | 4786.74 | -0.83 | 4837.25 | 0.22 | 13.09 | **4765.97** | -1.26 |
| 125c6d7v | 5325.28 | 5325.28 | 0.00 | 5466.55 | 2.65 | 9.18 | 5221.52 | -1.95 | 5295.95 | -0.55 | 12.89 | **5164.18** | -3.03 |
| 125c8d3v | 4553.28 | 4553.28 | 0.00 | 4677.51 | 2.73 | 10.23 | 4574.82 | 0.47 | 4621.98 | 1.51 | 14.98 | **4545.44** | -0.17 |
| 125c8d5v | 5045.65 | 5045.65 | 0.00 | 5115.35 | 1.38 | 9.64 | **4958.26** | -1.73 | 5139.14 | 1.85 | 13.38 | **4958.26** | -1.73 |
| 125c8d7v | 5416.96 | 5416.96 | 0.00 | 5450.87 | 0.63 | 9.34 | 5397.86 | -0.35 | 5473.96 | 1.05 | 13.38 | **5347.1** | -1.29 |
| 150c4d3v | **4049.48** | **4049.48** | 0.00 | 4050.08 | 0.01 | 9.71 | 4072.95 | 0.58 | 4172.94 | 3.05 | 21.84 | 4069.72 | 0.50 |
| 150c4d5v | 4638.72 | 4638.72 | 0.00 | 4705.63 | 1.44 | 8.19 | **4622.77** | -0.34 | 4666.79 | 0.61 | 19.11 | **4622.77** | -0.34 |
| 150c4d7v | 5176.5 | 5176.5 | 0.00 | 5243.96 | 1.30 | 8 | 5163.02 | -0.26 | 5205.56 | 0.56 | 19.06 | **5137.69** | -0.75 |
| 150c6d3v | **4057.09** | **4057.09** | 0.00 | 4063.34 | 0.15 | 9.96 | 4066.71 | 0.24 | 4116.11 | 1.45 | 22.07 | 4062.53 | 0.13 |
| 150c6d5v | **4872.08** | **4872.08** | 0.00 | 4898.39 | 0.54 | 10.23 | 4931.13 | 1.21 | 4989.97 | 2.42 | 21.16 | 4876.91 | 0.10 |
| 150c6d7v | 5768.29 | 5768.29 | 0.00 | 5916.88 | 2.58 | 10.73 | 5840.52 | 1.25 | 5883.53 | 2.00 | 20.4 | **5712.01** | -0.98 |
| 150c8d3v | 4653.9 | 4653.9 | 0.00 | 4737.16 | 1.79 | 10.18 | 4689.13 | 0.76 | 4823.95 | 3.65 | 22.67 | 4667.5 | 0.29 |
| 150c8d5v | 5113.77 | 5113.77 | 0.00 | 5169.84 | 1.10 | 11.62 | 5116.55 | 0.05 | 5200.19 | 1.69 | 19.6 | **5073.8** | -0.78 |
| 150c8d7v | 5665.23 | 5665.23 | 0.00 | 5665.27 | 0.00 | 12.01 | 5648.32 | -0.30 | 5693.24 | 0.49 | 19.67 | **5612.02** | -0.94 |
| 175c4d4v | **4706.76** | **4706.76** | 0.00 | 4782.13 | 1.60 | 21.74 | 4720.36 | 0.29 | 4781.93 | 1.60 | 28.69 | 4708.66 | 0.04 |
| 175c4d6v | **4835.64** | **4835.64** | 0.00 | 4960.33 | 2.58 | 23.01 | 4863.88 | 0.58 | 4956.47 | 2.50 | 26.71 | 4841.51 | 0.12 |
| 175c4d8v | 5943.28 | 5943.28 | 0.00 | 6034.04 | 1.53 | 18.4 | 5853.9 | -1.50 | 5934.35 | -0.15 | 27.35 | **5832.26** | -1.87 |
| 175c6d4v | 5025.51 | 5025.51 | 0.00 | 5108.08 | 1.64 | 21.51 | 5011.01 | -0.29 | 5120.82 | 1.90 | 29.28 | **5020.01** | -0.11 |
| 175c6d6v | 5431.34 | 5431.34 | 0.00 | 5437.14 | 0.11 | 22.54 | 5382.57 | -0.90 | 5483.57 | 0.96 | 27.43 | **5360.35** | -1.31 |
| 175c6d8v | 6090.01 | 6090.01 | 0.00 | 6167.31 | 1.27 | 25.81 | 6066.1 | -0.39 | 6156.06 | 1.08 | 27.97 | **6043.43** | -0.76 |
| 175c8d4v | 5878.58 | 5878.58 | 0.00 | 6031.02 | 2.59 | 24.9 | 5840.25 | -0.65 | 5954.97 | 1.30 | 29.83 | **5822.55** | -0.95 |
| 175c8d6v | 5989.63 | 5989.63 | 0.00 | 6157.32 | 2.80 | 25.21 | 5968.99 | -0.34 | 6123.9 | 2.24 | 27.78 | **5953.54** | -0.60 |
| 175c8d8v | 6943.63 | 6943.63 | 0.00 | 7075.23 | 1.90 | 26.7 | 6840.04 | -1.49 | 7054.85 | 1.60 | 27.98 | **6775.68** | -2.42 |
| Average | | | 0.00 | | 1.60 | 9.54 | | -0.05 | | 1.44 | 12.79 | | -0.45 |

* Note that this value corresponds to the best solution ever obtained with the final parameter setting

Table 6: Comparison of the performance of our VNS/TS heuristic on the MDVRPI instances proposed by Tarantilis et al. (2008) with the solutions of HGL. BKS denotes the previously best known solution. Gaps are calculated in dependence of BKS. Additionally, we provide the best solutions in $\overline{\text{VNS/TS}}$ that we ever obtained on the instances during our testing activities. $t(min)$ denotes the average runtime for each run.

|  | CPLEX |  |  |  |  | MCWS |  | DBCA |  | VNS/TS |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  | $m$ | $n$ | $L_{best}(S)$ | $m$ | $n$ | $L_{best}(S)$ | $\Delta(\%)$ | $L_{best}(S)$ | $\Delta(\%)$ | $m$ | $n$ | $L_{best}(S)$ | $t(min)$ | $\Delta(\%)$ |
| 20c3sU1 | **6** | 20 | **1797.49** | **6** | 20 | 1818.35 | 1.16 | **1797.51** | 0.00 | **6** | 20 | **1797.49** | 0.69 | 0.00 |
| 20c3sU2 | **6** | 20 | **1574.77** | **6** | 20 | 1614.15 | 2.50 | 1613.53 | 2.46 | **6** | 20 | **1574.77** | 0.64 | 0.00 |
| 20c3sU3 | **6** | 20 | **1704.48** | 7 | 20 | 1969.64 | 15.56 | 1964.57 | 15.26 | **6** | 20 | **1704.48** | 0.64 | 0.00 |
| 20c3sU4 | **5** | 20 | **1482** | 6 | 20 | 1508.41 | 1.78 | 1487.15 | 0.35 | **5** | 20 | **1482** | 0.65 | 0.00 |
| 20c3sU5 | 6 | 20 | **1689.37** | **5** | 20 | 1752.73 | 3.75 | 1752.73 | 3.75 | 6 | 20 | **1689.37** | 0.67 | 0.00 |
| 20c3sU6 | **6** | 20 | **1618.65** | **6** | 20 | 1668.16 | 3.06 | 1668.16 | 3.06 | **6** | 20 | **1618.65** | 0.67 | 0.00 |
| 20c3sU7 | **6** | 20 | **1713.66** | **6** | 20 | 1730.45 | 0.98 | 1730.45 | 0.98 | **6** | 20 | **1713.66** | 0.64 | 0.00 |
| 20c3sU8 | **6** | 20 | **1706.5** | **6** | 20 | 1718.67 | 0.71 | 1718.67 | 0.71 | **6** | 20 | **1706.5** | 0.67 | 0.00 |
| 20c3sU9 | **6** | 20 | **1708.81** | **6** | 20 | 1714.43 | 0.33 | 1714.43 | 0.33 | **6** | 20 | **1708.81** | 0.66 | 0.00 |
| 20c3sU10 | **4** | 20 | **1181.31** | 5 | 20 | 1309.52 | 10.85 | 1309.52 | 10.85 | **4** | 20 | **1181.31** | 0.64 | 0.00 |
| 20c3sC1 | **4** | 20 | **1173.57** | 5 | 20 | 1300.62 | 10.83 | 1300.62 | 10.83 | **4** | 20 | **1173.57** | 0.62 | 0.00 |
| 20c3sC2 | **5** | 19 | **1539.97** | **5** | 19 | 1553.53 | 0.88 | 1553.53 | 0.88 | 5 | 19 | **1539.97** | 0.58 | 0.00 |
| 20c3sC3 | **3** | 12 | **880.2** | 4 | 12 | 1083.12 | 23.05 | 1083.12 | 23.05 | **3** | 12 | **880.2** | 0.25 | 0.00 |
| 20c3sC4 | **4** | 18 | **1059.35** | 5 | 18 | 1135.9 | 7.23 | 1091.78 | 3.06 | **4** | 18 | **1059.35** | 0.53 | 0.00 |
| 20c3sC5 | 7 | 19 | - | **7** | 19 | 2190.68 |  | 2190.68 |  | 7 | 19 | **2156.01** | 0.6 |  |
| 20c3sC6 | **8** | 17 | **2758.17** | 9 | 17 | 2883.71 | 4.55 | 2883.71 | 4.55 | **8** | 17 | **2758.17** | 0.71 | 0.00 |
| 20c3sC7 | **4** | 6 | **1393.99** | 5 | 6 | 1701.4 | 22.05 | 1701.4 | 22.05 | **4** | 6 | **1393.99** | 0.18 | 0.00 |
| 20c3sC8 | **9** | 18 | **3139.72** | 10 | 18 | 3319.74 | 5.73 | 3319.74 | 5.73 | **9** | 18 | **3139.72** | 0.62 | 0.00 |
| 20c3sC9 | **6** | 19 | **1799.94** | **6** | 19 | 1811.05 | 0.62 | 1811.05 | 0.62 | 6 | 19 | **1799.94** | 0.6 | 0.00 |
| 20c3sC10 | **8** | 15 | - | **8** | 15 | 2648.84 |  | 2644.11 |  | 8 | 15 | **2583.42** | 0.45 |  |
| S1_2i6s | **6** | 20 | **1578.12** | **6** | 20 | 1614.15 | 2.28 | 1614.15 | 2.28 | 6 | 20 | **1578.12** | 0.71 | 0.00 |
| S1_4i6s | **5** | 20 | 1413.96 | **5** | 20 | 1561.3 | 10.42 | 1541.46 | 9.02 | 5 | 20 | **1397.27** | 0.75 | -1.18 |
| S1_6i6s | **5** | 20 | **1560.49** | 6 | 20 | 1616.2 | 3.57 | 1616.2 | 3.57 | 5 | 20 | **1560.49** | 0.73 | 0.00 |
| S1_8i6s | **6** | 20 | **1692.32** | **6** | 20 | 1902.51 | 12.42 | 1882.54 | 11.24 | 6 | 20 | **1692.32** | 0.74 | 0.00 |
| S1_10i6s | **4** | 20 | **1173.48** | 5 | 20 | 1309.52 | 11.59 | 1309.52 | 11.59 | **4** | 20 | **1173.48** | 0.71 | 0.00 |
| S2_2i6s | **6** | 20 | **1633.1** | **6** | 20 | 1645.8 | 0.78 | 1645.8 | 0.78 | 6 | 20 | **1633.1** | 0.75 | 0.00 |
| S2_4i6s | **5** | 19 | 1555.2 | 6 | 19 | **1505.06** | -3.22 | **1505.06** | -3.22 | 5 | 19 | 1532.96 | 0.88 | -1.43 |
| S2_6i6s | **7** | 20 | - | 10 | 20 | 3115.1 |  | 3115.1 |  | **7** | 20 | **2431.33** | 0.78 |  |
| S2_8i6s | **7** | 16 | **2158.35** | 9 | 16 | 2722.55 | 26.14 | 2722.55 | 26.14 | **7** | 16 | **2158.35** | 0.57 | 0.00 |
| S2_10i6s | **6** | *17* | - | **6** | 16 | 1995.62 |  | 1995.62 |  | 6 | *17* | **1958.46** | 0.61 |  |
| S1_4i2s | 6 | 20 | **1582.21** | **6** | 20 | **1582.2** | 0.00 | **1582.2** | 0.00 | 6 | 20 | **1582.21** | 0.63 | 0.00 |
| S1_4i4s | **5** | 20 | **1460.09** | 6 | 20 | 1580.52 | 8.25 | 1580.52 | 8.25 | **5** | 20 | **1460.09** | 0.68 | 0.00 |
| S1_4i6s | 5 | 20 | **1397.27** | **5** | 20 | 1561.29 | 11.74 | 1541.46 | 10.32 | 5 | 20 | **1397.27** | 0.75 | 0.00 |
| S1_4i8s | 6 | 20 | 1403.57 | **6** | 20 | 1561.29 | 11.24 | 1561.29 | 11.24 | 6 | 20 | **1397.27** | 0.82 | -0.45 |
| S1_4i10s | 5 | 20 | 1397.27 | **5** | 20 | 1536.04 | 9.93 | 1529.73 | 9.48 | 5 | 20 | **1396.02** | 0.85 | -0.09 |
| S2_4i2s | **4** | 18 | **1059.35** | 5 | 18 | 1135.89 | 7.23 | 1117.32 | 5.47 | **4** | 18 | **1059.35** | 0.51 | 0.00 |
| S2_4i4s | **5** | 19 | **1446.08** | 6 | 19 | 1522.72 | 5.30 | 1522.72 | 5.30 | **5** | 19 | **1446.08** | 0.6 | 0.00 |
| S2_4i6s | **5** | 20 | **1434.14** | 6 | 20 | 1786.21 | 24.55 | 1730.47 | 20.66 | **5** | 20 | **1434.14** | 0.69 | 0.00 |
| S2_4i8s | **5** | 20 | **1434.14** | 6 | 20 | 1786.21 | 24.55 | 1786.21 | 24.55 | **5** | 20 | **1434.14** | 0.75 | 0.00 |
| S2_4i10s | **5** | 20 | **1434.13** | 6 | 20 | 1783.63 | 24.37 | 1729.51 | 20.60 | **5** | 20 | **1434.13** | 0.78 | 0.00 |
| Average | 5.58 | 18.8 | 1575.98 | 6.13 | 18.78 | 1781.42 | 8.52 | 1774.15 | 7.94 | 5.58 | 18.8 | 1645.45 | 0.65 | -0.09 |

Table 7: Results on the small-sized G–VRP instances. Comparison of the solutions obtained by the MCWS and DBCA heuristics, the solutions determined by our CPLEX implementation and those of our VNS/TS. $L_{best}(S)$ denotes the best solution found in 10 runs, and $\Delta$ the gap to the best known solution (BKS). $t(min)$ reports the average computing time in minutes. We terminate CPLEX after 3 hours, so optimality is for none of the solutions guaranteed. Numbers in bold indicate the best solution found. Note that in some cases, our preprocessing identified a higher number of feasible customers (numbers in italic) than Erdogan and Miller-Hooks (2012).

in the first columns.

| | BKS | | | MCWS | | | | DBCA | | VNS/TS | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $m$ | $n$ | $L_{best}(S)$ | $m$ | $n$ | $L_{best}(S)$ | $\Delta(\%)$ | $L_{best}(S)$ | $\Delta(\%)$ | $m$ | $n$ | $L_{best}(S)$ | $t(min)$ | $\Delta(\%)$ |
| 111c_21 | **17** | 109 | **4797.15** | 20 | 109 | 5626.64 | 17.29 | 5626.64 | 17.29 | **17** | 109 | **4797.15** | 21.76 | 0.00 |
| 111c_22 | **17** | 109 | **4802.16** | 20 | 109 | 5610.57 | 16.83 | | | **17** | 109 | **4802.16** | 23.56 | 0.00 |
| 111c_24 | **17** | 109 | **4786.96** | 20 | 109 | 5412.48 | 13.07 | | | **17** | 109 | **4786.96** | 21.9 | 0.00 |
| 111c_26 | **17** | 109 | **4778.62** | 20 | 109 | 5408.38 | 13.18 | | | **17** | 109 | **4778.62** | 25.12 | 0.00 |
| 111c_28 | **17** | 109 | **4799.15** | 20 | 109 | 5331.93 | 11.10 | | | **17** | 109 | **4799.15** | 24.17 | 0.00 |
| 200c | **35** | *192* | **8963.46** | **35** | 190 | 10428.59 | 16.35 | 10413.59 | 16.18 | **35** | *192* | **8963.46** | 76.65 | 0.00 |
| 250c | **39** | *237* | **10800.18** | 41 | 235 | 11886.61 | 10.06 | 11886.61 | 10.06 | **39** | *237* | **10800.18** | 120.9 | 0.00 |
| 300c | **46** | *283* | **12594.77** | 49 | 281 | 14242.56 | 13.08 | 14229.92 | 12.98 | **46** | *283* | **12594.77** | 182.23 | 0.00 |
| 350c | **51** | 329 | **14323.02** | 57 | 329 | 16471.10 | 15.00 | 16460.30 | 14.92 | **51** | 329 | **14323.02** | 232.03 | 0.00 |
| 400c | **61** | 378 | **16850.21** | 67 | 378 | 19472.10 | 15.56 | 19099.04 | 13.35 | **61** | 378 | **16850.21** | 305.12 | 0.00 |
| 450c | **68** | 424 | **18521.23** | 75 | 424 | 21854.17 | 18.00 | 21854.19 | 18.00 | **68** | 424 | **18521.23** | 525.52 | 0.00 |
| 500c | **76** | 471 | **21170.9** | 84 | 471 | 24527.46 | 15.85 | 24517.08 | 15.81 | **76** | 471 | **21170.9** | 356.01 | 0.00 |
| Average | 38.42 | 238.25 | 10598.98 | 42.33 | 237.75 | 12189.38 | 14.61 | 15510.92 | 14.82 | 38.42 | 238.25 | 10598.98 | 159.58 | 0.00 |

Table 8: Results on the large-scale G–VRP instances. Comparison of the solutions obtained by the MCWS and DBCA heuristics and those of our VNS/TS. Better solutions are marked in bold; differences below 0.3 are neglected. $L_{best}(S)$ denotes the best solution found in 10 runs, and $\Delta$ the gap to the best known solution (BKS). $t(min)$ reports the average computing time in minutes. Numbers in bold indicate the best solution found. Note that in some cases, our preprocessing identified a higher number of feasible customers (numbers in italic) than Erdogan and Miller-Hooks (2012).

The results obtained by our VNS/TS heuristic on the large instance set is even more impressive. The solutions of MCWS and DBCA are on average almost 15% worse than our solutions. In addition, we require significantly less 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 instance, we obtain competitive results while requiring moderate computing times.

## 6 Conclusion

In this paper, we present a new vehicle routing problem for determining cost-optimal routes for electric vehicles. The E-VRPTW considers a limited vehicle and battery capacity and traveling along arcs consumes battery charge according to a constant consumption factor $r$. Vehicles have the possibility of visiting recharging stations along the route. The recharging time depends on the current battery charge on arrival at the station. Furthermore, customer time windows are incorporated into the E-VRPTW model in order to represent real-world requirements.

We develop a hybrid VNS/TS heuristic, which 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. Furthermore, we increase the diversification abilities of our method by implementing an acceptance criterion based on the Metropolis probability. In numerical studies performed on newly designed E-VRPTW benchmark instances, we demonstrate the positive effect of combining the two metaheuristics VNS and TS. Moreover, we solve benchmark instances of the related problems MDVRPI and G-VRP. Although our VNS/TS algorithm is not specifically tailored to solve those problems, it outperforms all competing algorithms on both G-VRP instance sets as well as on the large MDVRPI instance set. It is also worth mentioning that we found new best solutions for a large number of benchmark instances available for MDVRPI and G-VRP.

## References

A. Artmeier, J. Haselmayr, and M. Leucker, M.and Sachenbacher. The shortest path problem revisited: Optimal routing for electric vehicles. In *KI 2010: Advances in Artificial Intelligence, LNCS, Volume*

*6359*, pages 309–316. Springer, 2010.

R. Baldacci, A. Mingozzi, and R. Roberti. Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *European Journal of Operational Research*, 218(1): 1–6, 2012.

A. Boostani, R. Ghodsi, and A. K. Miab. Optimal location of compressed natural gas (CNG) refueling station using the arc demand coverage model. In *Proceedings of the 2010 Fourth Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation*, AMS '10, pages 193–198. IEEE Computer Society, 2010.

O. Bräysy. A reactive variable neighborhood search for the vehicle routing problem with time windows. *INFORMS Journal on Computing*, 15(4):347–368, 2003.

O. Bräysy and M. Gendreau. Vehicle routing problem with time windows, Part II: Metaheuristics. *Transportation Science*, 39(1):119–139, 2005a.

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

J.-F. Cordeau, G. Laporte, and A. Mercier. A unified tabu search heuristic for vehicle routing problems with time windows. *The Journal of the Operational Research Society*, 52(8):928–936, 2001.

B. Crevier, J.-F. Cordeau, and G. Laporte. The multi-depot vehicle routing problem with inter-depot routes. *European Journal of Operational Research*, 176(2):756–773, 2007.

G. B. Dantzig and J. H. Ramser. The truck dispatching problem. *Management Science*, 6:80–91, 1959.

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

European Comission. White Paper - Roadmap to a single european transport area – Towards a competitive and resource efficient transport system, 2011. URL `http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2011:0144:FIN:EN:PDF`.

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, 2011.

M. Gendreau and J. Y. Potvin. Tabu search. In M. Gendreau and J.Y. Potvin, editors, *Handbook of Metaheuristics*, pages 41–59. Springer, 2010.

M Gendreau and C. D. Tarantilis. Solving large-scale vehicle routing problems with time windows: The state-of-the-art. Technical report, CIRRELT-2010-04, 2010.

F. Glover and M. Laguna. *Tabu Search*. Kluwer Academic Publishers, 1997.

F. Gonçalves, S. R. Cardoso, Relvas S., and A. P. F. D Barbosa-Póvoa. Optimization of a distribution network using electric vehicles: A VRP problem. Technical report, CEG-IST, UTL, Lisboa, 2011.

V. C. Hemmelmayr, K. F. Doerner, and R. F. Hartl. A variable neighborhood search heuristic for periodic routing problems. *European Journal of Operational Research*, 195(2):791–802, 2009.

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

B.-I. Kim, S Kim, and S. Sahoo. Waste collection vehicle routing problem with time windows. *Computers & Operations Research*, 33(12):3624–3642, 2006.

G. Kindervater and M. Savelsbergh. Vehicle routing: Handling edge exchanges. In E. Aarts and J.K. Lenstra, editors, *Local Search in Combinatorial Optimization*, chapter 10, pages 337–360. John Wiley & Sons Ltd., 1997.

S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598): 671–680, 1983.

G. Laporte. Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416, 2009.

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

A. Mehrez and H. I. Stern. Optimal refueling strategies for a mixed-vehicle fleet. *Naval Research Logistics Quarterly*, 32(2):315–328, 1985.

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

A. A. Melkman, H. I. Stern, and A. Mehrez. Optimal refueling sequence for a mixed fleet with limited refuelings. *Naval Research Logistics Quarterly*, 33(4):759–762, 1986.

N. Mladenović and P. Hansen. Variable neighborhood search. *Computers & Operations Research*, 24(11): 1097–1100, 1997.

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

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

J.-Y. Potvin and J.-M. Rousseau. An exchange heuristic for routing problems with time windows. *Journal of the Operational Research Society*, 46(12):1433–1446, 1995.

E. Prescott-Gagnon, G. Desaulniers, and L.-M. Rousseau. A branch-and-price-based large neighborhood search algorithm for the vehicle routing problem with time windows. *Networks*, 54(4):190–204, 2009.

H. N. Psaraftis. k-Interchange procedures for local search in a precedence-constrained routing problems. *European Journal of Operational Research*, 13(4):391–402, 1983.

Y. Qiu, H. Liu, D. Wang, and X. Liu. Intelligent strategy on coordinated charging of PHEV with TOU price. In *Power and Energy Engineering Conference APPEEC 2011 AsiaPacific*, pages 1–5, 2011.

Y. Rochat and E. D. Taillard. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics*, 1(1):147–167, 1995.

M. W. P. Savelsbergh. Local search in routing problems with time windows. *Annals of Operations Research*, 4(1):285–305, 1985.

M. W. P. Savelsbergh. The vehicle routing problem with time windows: Minimizing route duration. *ORSA Journal on Computing*, 4(2):146–154, 1992.

M. Schneider, B. Sand, and A. Stenger. A note on the time travel approach for handling time windows in vehicle routing problems. Working Paper, BISOR, Technical University Kaiserslautern, 2012. URL `http://bisor.wiwi.uni-kl.de/fileadmin/ci/time_travel.pdf`.

M. M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265, 1987.

A. Stenger, D. Vigo, S. Enz, and M. Schwind. An adaptive variable neighborhood search algorithm for a vehicle routing problem arising in small package shipping. *Transportation Science*, page forthcoming, 2011.

C. D. Tarantilis, E. E. Zachariadis, and C. T. Kiranoudis. A hybrid guided local search for the vehicle-routing problem with intermediate replenishment facilities. *INFORMS Journal on Computing*, 20 (1):154–168, 2008.

P. M. Thompson and J. B. Orlin. Theory of cyclic transfers. Working Paper, Operations Research Center, MIT, Cambridge, Mass, 1989.

P. M. Thompson and H. N. Psaraftis. Cyclic transfer algorithms for multivehicle routing and scheduling problems. *Operations Research*, 41(5):935–946, 1993.

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

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

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

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