European Journal of Operational Research 250 (2016) 101–119

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

The vehicle-routing problem with time windows and driver-specific times

Michael Schneider*

Logistics Planning and Information Systems, TU Darmstadt, 64289 Darmstadt, Hessen, Germany

ARTICLE INFO

Article history:
Received 6 August 2014
Accepted 8 September 2015
Available online 16 September 2015

Keywords:
Vehicle routing
Time windows
Driver-specific times
Routing consistency
Metaheuristics

ABSTRACT

This paper proposes a tabu search algorithm for the vehicle-routing problem with time windows and driver-specific times (VRPTWDST), a variant of the classical VRPTW that uses driver-specific travel and service times to model the familiarity of the different drivers with the customers to visit. We carry out a systematic investigation of the problem on a comprehensive set of newly generated benchmark instances. We find that consideration of driver knowledge in the route planning clearly improves the efficiency of vehicle routes, an effect that intensifies for higher familiarity levels of the drivers. Increased benefits are produced if the familiar customers of drivers are geographically contiguous. Moreover, a higher number of drivers that are familiar with the same (larger) region provides higher benefits compared to a scenario where each driver is only familiar with a dedicated (smaller) region. Finally, our tabu search is able to prove its performance on the Solomon test instances of the closely related VRPTW, yielding high-quality solutions in short time.

1. Introduction

Small package shipping (SPS) is a fast-growing market strongly driven by the expanding e-commerce sector. Besides the highly competitive market situation, rising fuel and labor costs constantly decrease the players' profit margins per delivered package. In general, SPS companies perform last-mile deliveries from a set of local depots (see, e.g., Zhong, Hall, & Dessouky, 2007) and the associated pickup and delivery costs are estimated to amount to 35–60 percent of the total transportation cost (Wasner & Zäpfel, 2004). To render their local pickup and delivery operations profitable and to gain advantage over their competitors, SPS companies pay more and more attention to effective workforce management practices (Smilowitz, Nowak, & Jiang, 2012). An effective utilization of drivers can be achieved by employing drivers to serve the customers and regions that they are most familiar with. For example, experienced drivers use shortcuts, know about traffic light intervals, anticipate road or traffic problems, find parking space more easily and know alternative delivery possibilities in case a customer is absent. This leads to reduced travel and service times. In addition, a high service consistency is achieved, i.e., customers are often served by the same driver, leading to increased customer satisfaction and a stronger bond between company and customer.

To effectively manage local delivery tasks, operations research techniques are applied, where route planning is generally represented as vehicle-routing problem (VRP; Toth & Vigo, 2014). Because customers with time-definite delivery requirements are very common in the SPS sector and constitute up to 60 percent of the total delivery volume (Campbell & Thomas, 2009), the VRP with time windows (VRPTW) lies at the heart of the local delivery operations of an SPS company. It incorporates the most important practical constraints, namely a limited freight capacity of vehicles and the requirement that each customer can only be visited within a given time window (for a recent survey on the VRPTW, see, e.g., Desaulniers, Røpke, & Madsen, 2014). To be more precise, the VRPTW calls for the determination of a cost-minimal set of routes carried out by a set of identical vehicles located at a single depot. Each route starts and ends at the depot within a given scheduling horizon and the cumulative demand of the customers visited on a route does not exceed vehicle capacity. Each customer is served by exactly one vehicle within its given time window. The VRPTW is an NP-hard problem of which only small to medium-sized instances can be solved within acceptable computation times by means of exact solution methods (Baldacci, Mingozzi, & Roberti, 2012). Consequently, a huge number of metaheuristic solution methods have been proposed (see Bräysy & Gendreau, 2005b; Gendreau, Potvin, Bräysy, Hasle, & Løkketangen, 2008; Gendreau & Tarantilis, 2010).

This paper addresses a variant of the VRPTW that incorporates the workforce management decision of employing drivers in the most beneficial manner, i.e., the route planning model aims

* Tel.: +4915209818017.
E-mail address: schneider@bwl.tu-darmstadt.de

http://dx.doi.org/10.1016/j.ejor.2015.09.015
0377-2217/© 2015 Elsevier B.V. and Association of European Operational Research Societies (EURO) within the International Federation of Operational Research Societies (IFORS). All rights reserved.

at generating efficient routes based on different extents of driver knowledge. The familiarity of the different drivers with the customers to visit is modeled by means of driver-specific travel and service times. Thus, the resulting VRP with time windows and driver-specific times (VRPTWDST) additionally asks for the best assignment of drivers to their familiar customers (regions), which is a prerequisite for generating overall efficient vehicle routes. The driver specific times are expressed by so called familiarity factors, which give the reduction of the time requirement of the familiar driver as ratio of the time requirement of a completely unfamiliar driver. For example, a familiarity factor of 0.1 states that the time of a familiar driver reduces to 90 percent of the original time of an unfamiliar driver.

Note that vehicle-dependent travel times have already been proposed some time ago (see, e.g., Cordeau, Desaulniers, Desrosiers, Solomon, & Soumis, 2002) and are sometimes included in heterogeneous fleet VRP (HFVRP) models. In this sense, VRPTWDST may be seen as a variant of the HFVRP with time windows (HFVRPTW; Liu & Shen, 1999), in which the capacity and cost of all vehicles are identical and travel costs are vehicle-specific. We refrain from this nomenclature for two reasons. First, existing works on HFVRPTW have their main focus on the different characteristics of the employed vehicles (vehicle capacity and cost) while in our problem a homogeneous vehicle fleet is considered. Second, the vast majority of HFVRPTW papers does not cover vehicle-specific times, and the regularly used LS168 benchmark of Liu and Shen (1999) does not include them. The few works that consider vehicle-specific times only address them in the presented mathematical models but not in the algorithm design or in the numerical studies (for details, see the literature review in Section 2).

The idea of using driver-specific cost to represent different levels of driver familiarity was originally introduced by Zhong et al. (2007) and is also used in Schneider, Doppstadt, Sand, Stenger, and Schwind (2010a) and Schneider, Doppstadt, Stenger, and Schwind (2010b). However, to the best of our knowledge, no work systematically investigating driver-specific times has been presented yet. The contribution of our paper lies in this area. Because VRPTWDST extends the VRPTW, the high complexity of the problem renders exact solution methods inadequate for solving realistically sized problem instances within fast computation times. Therefore, we develop a tabu search (TS), called TS-DST, to address the problem. We generate a comprehensive set of VRPTWDST test instances and conduct numerical experiments to investigate the effect of (i) different geographical distributions of the familiar customers, (ii) different levels of driver familiarity (expressed by different familiarity factors), and (iii) different distributions of familiarity among the drivers, on the resulting routing quality. We further use the well-known Solomon benchmark of the closely related VRPTW to prove the general ability of our algorithm to address hard-to-solve routing problems with time windows.

Many SPS routing approaches generate consistent vehicle routes based on the fixed assignment of service territories to drivers (see, e.g., Schneider, Stenger, Schwahn, & Vigo, 2014; Zhong et al., 2007) or based on template routes (see, e.g., Groër, Golden, & Wasil, 2009; Sungur, Ren, Ordóñez, Dessouky, & Zhong, 2010). In this way, driver learning is promoted, and once drivers are familiar with their customers/region, they are basically always used for serving their familiar customers. Contrary to such approaches, VRPTWDST works on the assumption that driver learning has already taken place in the past and different extents of familiarity exist among the drivers. Then, VRPTWDST aims at employing these drivers in the best possible fashion, which as a side effect will also use them for serving their familiar customers because of shorter travel and service time. Thus, delivery consistency is encouraged without enforcing it by means of fixed service territories or template routes, and routing flexibility is maintained. The latter is vital for generating efficient routes, especially if time windows are considered, because then flexibility is not only needed to achieve distance-efficient route configurations but also to fulfill time requirements (Schneider, Stenger, Schwahn, & Vigo, 2014).

Note that the VRPTWDST model and our solution method are not suitable to provide insights about the optimal tradeoff between routing consistency and routing efficiency, because VRPTWDST does not consider the costs for establishing the familiarity of the drivers, e.g., in the form of reduced routing efficiency or route feasibility (see Haughton, 2008; Smilowitz et al., 2012; Schneider, Stenger, Schwahn, & Vigo, 2014). However, VRPTWDST might be used to also consider the learning phase of drivers if applied in conjunction with a learning model that describes the reduction of travel and service times in dependence of the number of visits to each customer. This presents an interesting opportunity for future research. Finally, note that the contribution of this paper is not restricted to the SPS industry but is relevant for all routing operations in which (i) drivers exhibit different degrees of familiarity with regards to the customers, and (ii) where exploiting this familiarity is worthwhile.

The paper is organized as follows. Section 2 gives an overview of the related literature. Section 3 provides the mathematical formulation of the VRPTWDST. TS-DST is detailed in Section 4. The performance of the proposed method and the effects of driver-specific times are investigated in extensive numerical studies in Section 5. Finally, we give a short summary and conclusions in Section 6.

2. Literature

As described above, the VRPTWDST is closely related to the VRPTW and the HFVRP. Moreover, approaches addressing the familiarity of drivers with customers and workforce management in vehicle routing are of relevance. An overview of the literature concerning these topics is presented in the following.

The VRPTW is one of the most important and widely studied variants of the classical VRP, with hundreds of published papers (Gendreau & Tarantilis, 2010). Cordeau et al. (2002) and Desaulniers et al. (2014) provide comprehensive overviews of exact and heuristic methods for the VRPTW. A detailed description of the heuristic literature is given in the two-part review of Bräysy and Gendreau (2005a, 2005b), which is updated in Gendreau et al. (2008). Gendreau and Tarantilis (2010) provide an overview and analysis of advanced heuristics for solving large-scale VRPTW. A recent survey on exact algorithms is given by Baldacci et al. (2012). The best-performing metaheuristics for solving the VRPTW currently are memetic algorithms (Nagata, Bräysy, & Dullaert, 2010; Vidal, Crainic, Gendreau, & Prins, 2013), large neighborhood search (Bent & Hentenryck, 2004; Pisinger & Røpke, 2007; Prescott-Gagnon, Desaulniers, & Rousseau, 2009), iterated local search (Hashimoto, Yagiura, & Ibaraki, 2008), and multi-start local search (Lim & Zhang, 2007).

In the strand of HFVRP (for surveys, see Baldacci, Battarra, & Vigo, 2008; Irnich, Schneider, & Vigo, 2014), the following problems similar to the VRPTWDST have been studied. Ferland and Michelon (1988) address a vehicle scheduling problem accounting for vehicles of different types with varying travel and service times. They show how to extend some heuristic and exact methods for the standard vehicle scheduling problem to be able to deal with several vehicle types. No computational experiments are reported. Dondo and Cerdá (2006) study a dynamic VRPTW and later a multi-depot VRPTW with heterogeneous fleet (Dondo & Cerdá, 2007). Both problem formulations account for vehicle-specific times, however, all tests are conducted on problem instances with identical times for all vehicles. Goel and Gruhn (2008) study a rich HFVRP with time windows, vehicle-specific travel times, travel costs and capacities including several additional practical requirements like, e.g., a load acceptance decision. The authors propose a reduced variable neighborhood search and a large neighborhood search for the problem and compare the algorithms on a set of self-generated instances, which do not feature vehicle-specific times. As described in Section 1, the vast majority of methods proposed for HFVRPTW do not consider vehicle-specific times (see, e.g., Dell'Amico, Monaci, Pagani, & Vigo, 2007; Bräysy, Porkka, Dullaert, Repoussis, & Tarantilis, 2009; Repoussis & Tarantilis, 2010; Bettinelli, Ceselli, & Righini, 2011).

Approaches integrating workforce management aspects into routing models are often applied in the context of SPS. To find a tradeoff between achieving service consistency and maintaining routing flexibility, some recent scientific approaches present VRP models that explicitly integrate consistency requirements and are solved without fixing service territories (see Groër et al., 2009; Sungur et al., 2010; Smilowitz et al., 2012; Kovacs, Golden, Hartl, & Parragh, 2014a). On the other hand, approaches based on fixed territories are rendered more flexible, e.g., by excluding a percentage of customers from being assigned to fixed territories as done at UPS and described in Zhong et al. (2007). Here, the authors also integrate driver knowledge aspects by means of driver-specific costs. Their cell routing heuristic estimates the cost of inserting a cell into a specific driver's route by means of a learn factor that describes the driver's performance level for the considered cell. We refer the reader to Kovacs, Hartl, Parragh, and Golden (2014b) for a comprehensive survey of VRPs with consistency considerations.

One important issue when investigating VRPs with driver-specific times is the above described familiarity factor, i.e., the time reduction gained by a familiar driver in comparison to a completely unfamiliar driver. To the best of our knowledge, no empirical quantification of the learning effect on travel and service times is available. The values used in this paper are motivated from a personal communication with an industry partner from the SPS industry, who estimated the effect at a 40 percent time reduction for the familiar driver. The benchmark instances generated in this work use a maximal familiarity factor of 0.5, because we consider this value adequate for covering the maximal learning case.

3. Problem definition

In this section, we present a mathematical formulation of the VRPTWDST. Note that the VRPTWDST is a special case of the general VRP for which a model is given in Goel and Gruhn (2008).

Let $V = \{1, \ldots, N\}$ denote the set of N customers. Vertices 0 and $N+1$ denote instances of the depot. Further, let the indices 0 and $N+1$ indicate that a set contains the respective instance of the depot, e.g., $V_0 = V \cup \{0\}$. VRPTWDST is defined on a complete directed graph $G = (V_{0,N+1}, A)$, with $A = \{(i, j) \mid i, j \in V_{0,N+1}, i \neq j\}$ denoting the arc set. Identical vehicles with a maximal capacity of C are assumed to be stationed at the depot. Thus, a set K of driver/vehicle pairs is available that can uniquely be identified by the driver as is done in the following. With each vertex $i \in V_{0,N+1}$ are associated a nonnegative demand q_i ($q_0 = q_{N+1} = 0$), a time window $[e_i, l_i]$ (the time window of the depot $[e_0, l_0] = [e_{N+1}, l_{N+1}]$ corresponds to the scheduling horizon of the problem), and nonnegative service times s_{ik} that depend on the driver $k \in K$ visiting the vertex ($s_{0k} = s_{N+1k} = 0, k \in K$). Associated with each arc (i, j) is a distance d_{ij} and travel times t_{ijk} which denote the time driver k takes to travel the arc. The binary decision variable x_{ijk} equals 1 if driver k visits vertex j after i, and 0 otherwise. Variable τ_{ik} specifies the start of service of driver k at vertex i.

We study hierarchical objective functions for the VRPTWDST, first minimizing the number of vehicles and only considering a secondary objective in case of ties. We independently address two alternative secondary objectives: (i) minimize traveled distance, and (iii) minimize working duration, which are detailed in the following. Using a hierarchical objective function is inspired by heuristic works on the closely related VRPTW, however, the vast majority of works on VRPTW only consider the minimization of traveled distance as secondary objective.

1. Minimize traveled distance (TD):

$$\min \sum_{k \in K} \sum_{(i,j) \in A} d_{ij} x_{ijk}$$

Driver-specific times only indirectly influence TD, and potentially different routes with respect to the time window constraints are possible. This can lead to a reduced number of vehicles (because a higher number of customers can be served on routes) and to reduced traveled distances. Thus, this objective is suitable to analyze the benefits that can be achieved by being able to better meet time windows.

2. Minimize working duration (WD):

$$\min \sum_{k \in K} (\tau_{N+1k} - \tau_{0k})$$

WD is defined as the difference between the start and end time of the routes. It can alternatively be described as the sum of travel, service and waiting times of a solution. In addition to the benefits that can be gained by being more flexible with regards to time windows, this objective includes the time gains achieved by using arcs with reduced travel times and visiting customers with reduced service times. Here, the driver-specific times influence the objective function value in a direct fashion. This objective is interesting to study because in general payment of drivers is based on this value.

The constraints of the VRPTWDST are as follows:

$$\sum_{k \in K} \sum_{j \in V_{N+1}} x_{ijk} = 1 \quad \forall i \in V \tag{1}$$

$$\sum_{j \in V_{N+1}} x_{0jk} = 1 \quad \forall k \in K \tag{2}$$

$$\sum_{i \in V_0} x_{ijk} - \sum_{i \in V_{N+1}} x_{jik} = 0 \quad \forall j \in V, k \in K \tag{3}$$

$$x_{ijk}(\tau_{ik} + s_{ik} + t_{ijk} - \tau_{jk}) \leq 0 \quad \forall (i, j) \in A, k \in K \tag{4}$$

$$e_i \leq \tau_{ik} \leq l_i \quad \forall i \in V_{0,N+1}, k \in K \tag{5}$$

$$\sum_{(i,j) \in A} q_i x_{ijk} \leq C \quad \forall k \in K \tag{6}$$

$$x_{ijk} \in \{0, 1\} \quad \forall (i, j) \in A, k \in K \tag{7}$$

Constraints (1) state that each customer is visited by exactly one vehicle. Constraints (2) require each vehicle to start from the depot. Flow conservation constraints are given in (3). Time window and capacity restrictions are enforced by constraints (4)–(6). Binary variables are defined in constraints (7).

4. Tabu search for the VRPTWDST

TS is a local-search-based metaheuristic whose principles were first proposed in Glover (1986). In each iteration, the search considers the neighborhood $\mathcal{N}(S)$ of the current solution S. In general, the neighborhood is implicitly defined by means of neighborhood operators, which describe modifications to the current solution. In order to be able to escape from low-quality local optima, TS selects the best neighbor of the current solution at each iteration, even if it is of lower quality. In addition, recently visited solutions or attributes of the solutions are stored in a tabu list and are prohibited for some iterations called the tabu tenure. In this way, short term cycling of the search process is avoided. TS has successfully been applied to various combinatorial optimization problems providing near-optimal solutions in reasonable computing times. For a detailed description of TS and its extensions, we refer to Gendreau and Potvin (2010).

Our TS-DST for addressing the VRPTWDST applies some ideas similar to the ones presented in the TS methods of Toth and Vigo

```
initialize($tabuList, penaltyWeights$)
$S \leftarrow$ generateInitialSolution()
$S^* \leftarrow S$
$probabilisticPhase \leftarrow$ **false**
**repeat**
    **if** $probabilisticPhase$ **then**
        $S \leftarrow$ selectRandomOfBest($\mathcal{N}(S) \setminus tabuList, \xi$)
    **else**
        $S \leftarrow$ selectBestOf($\mathcal{N}(S) \setminus tabuList$)
    **end if**
    update($tabuList, S^*, penaltyWeights, probabilisticPhase$)
    **if** specific diversification criteria fulfilled **then**
        applyFurtherDiversification()
    **end if**
**until** termination criteria satisfied
**return** $S^*$
```

Fig. 1. Tabu search for the VRPTWDST in pseudocode.

(2003), Cordeau, Gendreau, and Laporte (1997), and Cordeau, Laporte, and Mercier (2001). An overview of the algorithm is given in pseudocode in Fig. 1. After eliminating infeasible arcs in a preprocessing step as described in Section 4.1, the initial solution is generated with an adapted version of the I1 heuristic presented by Solomon (1987) (Section 4.2). The initial solution and the solutions during the TS phase are allowed to be infeasible. They are evaluated by means of a generalized cost function that penalizes infeasible solutions based on the amount of capacity and time window violation (Section 4.3). In each iteration of TS-DST, a set of neighborhood operators are applied to the current solution S to generate the composite neighborhood $\mathcal{N}(S)$ (Section 4.4).

Each possible move is evaluated and the best non-tabu move is performed. However, if the algorithm carries out a so-called probabilistic phase for diversification, a random of the best ξ non-tabu moves is selected. A move is superior if it either reduces the number of employed vehicles or, in case of ties, has a better objective function value according to the generalized cost function described in Section 4.3. After executing the move, the tabu list, the best overall solution and the penalties are updated and the algorithm decides whether a probabilistic phase is started, continued or ended as described in Section 4.5. Besides probabilistic phases, we apply further search diversification techniques like, e.g., shaking the solution. The respective techniques, described in Section 4.5, are applied if the search has not improved the best found solution for a certain number of iterations. Finally, the termination criterion of the search depends on the mode of the TS-DST run, i.e., whether it is searching for a feasible solution with a given number of vehicles or minimizing the secondary objective (see Section 4.6 for details).

4.1. Preprocessing step

As often done for VRPs, we use a preprocessing step to remove those arcs from the solution graph that cannot belong to a feasible solution. An arc can either be infeasible due to time window or due to capacity constraints. To identify these arcs, we use the following rules and remove all arcs $(v, w) \in A$ for which one of the following conditions holds:

$$q_v + q_w \geq C \tag{8}$$

$$\min_{k \in K} (e_v + s_{vk} + t_{vwk}) \geq l_w \tag{9}$$

$$\min_{k \in K} (e_v + s_{vk} + t_{vwk} + s_{wk} + t_{wN+1k}) \geq l_{N+1} \tag{10}$$

Eq. (8) describes capacity violations that occur if the cumulative demand of two customers exceeds the vehicle capacity. Eqs. (9) and (10) adapt well-known preprocessing steps based on time window violations to VRPTWDST. Eq. (9) states that an arc can be removed if even the driver with the minimum travel and service time on this arc is not able to reach w within the given time window when starting service at v at the earliest possible time. Eq. (10) is similar but considers the depot deadline. If the driver with the minimum travel and service times is not able to return to the depot in time after consecutively visiting customers v and w, the arc (v, w) is infeasible.

```
$S \leftarrow \emptyset$
$V_{unrouted} \leftarrow V$
**while** $m(S) < m$ **and** $V_{unrouted} \neq \emptyset$ **do**
    $v_{seed} \leftarrow$ farthestFromDepot($V_{unrouted}$)
    $V_{unrouted} \leftarrow V_{unrouted} \setminus \{v_{seed}\}$
    $r \leftarrow \langle v_0, v_{seed}, v_{n+1} \rangle$
    **repeat**
        **for all** $v \in V_{unrouted}$ **do**
            $p^*(v) \leftarrow$ bestPosition($c_1, v, r$)
        **end for**
        $v^* \leftarrow$ bestCustomer($c_2, p^*, r$)
        $r \leftarrow$ insertCustomer($r, v^*, p^*$)
        $V_{unrouted} \leftarrow V_{unrouted} \setminus \{v^*\}$
    **until** no feasible insertion into $r$ possible
    $S \leftarrow$ addRoute($S, r$)
**end while**
**if** $V_{unrouted} \neq \emptyset$ **then**
    $S \leftarrow$ insertCustomers($S, V_{unrouted}, f_{gen}$)
**end if**
**return** $S$
```

Fig. 2. Generation of initial VRPTWDST solution in pseudocode.

Because the majority of the overall run-time of TS-DST is spent on the evaluation of local search moves, the impact of the preprocessing step on the run-time of the algorithm is determined by the percentage of arcs that can be eliminated in the problem instances. In Appendix A, we report the respective values for the newly generated VRPTWDST benchmark instances described in Section 5.2.

4.2. Generation of initial solution

To generate an initial solution, we use a slight modification of the insertion heuristic I1 introduced by Solomon (1987). Our algorithm creates a potentially infeasible solution with a given number of vehicles m instead of a feasible solution with an open number of vehicles as in the original version. We define a vehicle route r as a sequence of customers $\langle v_0, v_1, \ldots, v_n, v_{n+1} \rangle$, with v_0 and v_{n+1} representing the depot. A solution S to a VRPTWDST instance is defined as a set containing $m(S)$ routes, i.e., $S = \{r_k, k = 1, \ldots, m(S)\}$. The initialization procedure in pseudocode is shown in Fig. 2.

To be able to use the approach in the presence of driver-specific times, we calculate average travel and service times over the different drivers, thus transforming a VRPTWDST problem instance into a VRPTW instance. For the secondary objective of minimizing working duration, we solve an assignment problem afterwards (using IBM ILOG CPLEX Optimizer 12.1) to find the cost-optimal assignment of available drivers to the routes of the initial solution.

Our insertion heuristic sequentially opens routes and fills them with customers according to specific criteria until no more customers can be feasibly inserted. To open a new route, the procedure selects the unrouted customer farthest from the depot v_{seed} and generates a new route $r = \langle v_0, v_{\text{seed}}, v_{n+1} \rangle$. In the next step, for each of the still unrouted customers $v \in V_{\text{unrouted}}$ the best insertion position $p^*(v)$ between two consecutive customers in r is determined according to criterion c_1 in Eq. (11):

$$p^*(v) = \underset{p=1,\ldots,n}{\arg\min}\,(c_1(v_{p-1}, v, v_p)) \tag{11}$$

$$c_1(x, y, z) = \epsilon_1 \cdot c_{11}(x, y, z) + \epsilon_2 \cdot c_{12}(x, y, z) \tag{12}$$

$$c_{11}(x, y, z) = d_{xy} + d_{yz} - \iota \cdot d_{xz} \tag{13}$$

$$c_{12}(x, y, z) = a_z - a_z^{\text{old}} \tag{14}$$

$$\text{with} \quad \iota \geq 0, \quad \epsilon_1 + \epsilon_2 = 1, \quad \epsilon_1 \geq 0, \quad \epsilon_2 \geq 0$$

Eq. (12) considers the increase in traveled distance due to the insertion of the customer (calculated in Eq. (13)) and the increase of the arrival time at the successive customer (given in Eq. (14), here a_z^{old} denotes the arrival time before the insertion of the customer, a_z after the insertion). ϵ_1, ϵ_2 and ι are parameters for evaluating the cost of insertion of a customer.

Subsequently, the customer v^* that can be feasibly inserted and that maximizes Solomon's c_2 criterion is added to the route at its best insertion position (parameter ν is used for weighting the distance from the depot):

$$\begin{aligned} v^* &= \underset{v \in V_{\text{unrouted}} \wedge \text{feasibleInsert}(v)}{\arg\max} \left(c_2(v_{p^*(v)-1}, v, v_{p^*(v)})\right) \\ c_2(x, y, z) &= \nu \cdot d_{0y} - c_1(x, y, z) \\ \nu &\geq 0 \end{aligned}$$

These steps are repeated as long as a customer can be inserted into the route without violating constraints. If this is no longer possible, the route is closed, added to the solution S and the procedure continues with the next route until either m routes are created or each customer is routed. After m routes have been created, any customers that are still unrouted are added to the routes at the best insertion position according to the generalized cost function f_{gen} described in the following section.

4.3. Generalized cost function and penalty determination

We permit capacity and time window violations during the initialization and improvement phase using the generalized cost function $f_{\text{gen}}(S) = f(S) + \alpha \cdot P_c(S) + \beta \cdot P_{\text{tw}}(S)$, where $f(S)$ denotes the original objective function value of solution S, i.e., traveled distance or working duration, and $P_c(S)$ and $P_{\text{tw}}(S)$ are the total capacity and time window violations of S, which are weighted by the penalty factors α and β. The latter are dynamically updated during the search to allow for more diversification if the current solution is feasible and, in the contrary case, to guide the algorithm to a feasible solution (see, e.g., Gendreau, Hertz, & Laporte, 1994). The time window or capacity violation penalty factor is multiplied by a factor δ after the respective constraint is violated for η_{penalty} iterations. If the constraint is met for η_{penalty} iterations, the respective penalty factor is divided by δ. The penalty factors are restricted to the interval $[\alpha_{\min}, \alpha_{\max}]$ and $[\beta_{\min}, \beta_{\max}]$ respectively.

TS-DST uses the neighborhood operators Relocate, Exchange and 2-opt* (see Section 4.4). Consequently, the capacity penalty $P_c(S)$ of a solution S can be computed in $\mathcal{O}(1)$ time (see, e.g. Kindervater & Savelsbergh, 1997). To determine time window penalties, we adapt the time travel approach, proposed by Nagata et al. (2010) for computing time window violations for the VRPTW, to the VRPTWDST. Summarizing this technique, if a time window violation occurs at a certain customer, a penalty of the actual arrival time minus the latest feasible arrival time is assigned to the route. However, the calculation of violations for the following customers is executed as if a travel back in time to the latest feasible arrival time had taken place. In this way, none of the subsequent customer visits are affected by the actual delay at the customer. This approach has two important advantages compared to the traditional approach of propagating time window violations along the route. First, it allows to evaluate potential time window violations for inter-route moves in constant time $\mathcal{O}(1)$ for the conventional neighborhood operators, which significantly speeds up the calculation compared to the traditional approach. Second, the approach penalizes time window violations only at the customers where they originate and feasible parts of routes are no longer penalized by earlier time window violations.

Empirical investigations on the Solomon VRPTW instances have shown that the time travel approach provides superior solution quality compared to traditional time window handling techniques (see Vidal, Crainic, Gendreau, & Prins, 2015). Note that this may clearly be different for VRPTW instances with different characteristics or other VRPs with time window constraints like the VRPTWDST. Another possible disadvantage of the time travel approach is that the amount of time travel is not limited, i.e., it is possible to travel back in time to an extent that the adjusted start time of service at a violating customer lies before the start time of service at the preceding customer. However, the results in Vidal et al. (2015) suggest that this drawback has only minor influence on solution quality.

We use a straightforward adaption of the time travel approach: we determine all time window handling related variables of a route based on the driver-specific times of the driver currently assigned to the route. In this way, penalties for the Relocate and Exchange operator can be computed in constant time by applying the original rule for merging partial paths given in Nagata et al. (2010) and the corrected rule for inserting a vertex between partial paths presented in Schneider, Sand, and Stenger (2013). The reason is that for Relocate and Exchange both partial paths "belong" to the same driver and thus all time window handling related variables are available for the partial paths. However, this is not true if the rule for merging partial paths is applied for calculating the time window penalty of a 2-opt* move. Here, the first and the second partial path to be merged belong to different routes, which are carried out by different drivers. Therefore, the variables for the second partial path are not adequate for the driver performing the first partial path and vice versa and thus recalculations are necessary.

4.4. Neighborhood generation and tabu list

TS-DST combines several neighborhood operators to build a composite neighborhood:

Relocate is used for both inter and intra-route moves. The operator removes one vertex from a route and inserts it into another route at an arbitrary position (inter-route) or into the same route at a different position (intra-route), see Savelsbergh (1992).

Exchange is also used in inter and intra-route fashion and swaps the position of two vertices (Savelsbergh, 1992).

2-opt* is only used for inter-route moves. The operator removes one arc from each route and reconnects the first part of the first route with the second part of the second route and vice versa (Potvin & Rousseau, 1995).

The neighborhood in each iteration of TS-DST is generated by applying the above neighborhood operators for each arc of the list of generator arcs. Using arcs as move generators was originally introduced in Toth and Vigo (2003). The so-called generator arc (v, w) modifies the solution in a manner that after application of the move, vertex v is followed by vertex w. In this way, all other arcs to be removed and added are specified for a given operator. A more detailed description of the utilized neighborhood operators and their implementation via generator arcs can be found in Toth and Vigo (2003).

Concerning the implementation of the tabu list, it is well known that saving complete solutions in a tabu list is inefficient concerning storage and comparison effort. Therefore, attributes of a solution like, e.g., the presence of a certain object in the solution, or of a solution transition, e.g., the exchange of an object by another object, are stored, and moves are tabu if they induce these attributes. This also leads to a more extensive exploration of the search space as it helps the search to move away from previously visited parts of the search space. We follow the approach of Toth and Vigo (2003) and consider a move tabu if it reinserts an arc that was removed in one of the previous ϑ iterations, where ϑ denotes the tabu tenure. It is set to a random value in $[\vartheta_{\min}, \vartheta_{\max}]$ as proposed by Gendreau et al. (1994). As aspiration criterion, we allow the execution of a tabu move if it leads to a new overall best solution.

4.5. Diversification methods

Local-search-based heuristics tend to be too local in the sense that the search focuses on small regions of the search space. Therefore, the incorporation of diversification mechanisms, which aim at guiding the heuristic into previously unvisited regions of the solution space, is crucial in the design of efficient solution methods for complex combinatorial optimization problems like the VRPTWDST (Soriano & Gendreau, 1996). We apply several such mechanisms in TS-DST, which are described in the following.

4.5.1. Continuous diversification

Continuous diversification constantly tries to guide the search into previously unexplored regions of the solution space by adding a diversification penalty $P_{\text{div}}(S)$ to the cost of a solution S. The diversification penalty used in TS-DST is similar to the one given in Cordeau et al. (2001) but has to be defined for our two different objective functions. If a move leads to an inferior solution S', a diversification penalty $P_{\text{div}}(S')$ is added, which is computed as follows:

$$P_{\text{div}}(S') = \lambda_{\text{div}} \cdot f(S') \cdot \sqrt{|V|\, m(S')} \cdot \sum_{(v,k)\in B(S')} \varrho_{vk},$$

with the attribute set $B(S') = \{(v,k) : \text{customer } v \text{ is served by driver } k \text{ in solution } S'\}$, the occurrence frequency ϱ_{vk} of attribute (v,k), the diversification factor λ_{div} and the secondary objective function f.

4.5.2. Probabilistic phase

To further diversify the search, we use probabilistic phases, in which not the best solution is chosen in each iteration but a random one is picked from the list of the best ξ solutions. Once started, a probabilistic phase runs for at most η_{endProb} iterations or is stopped if a new best overall solution is found. The probabilistic phase is initialized as soon as one of the following two criteria is satisfied:

1. If the search has not improved the best overall solution S^* for $\eta_{\text{startProb}}$ iterations.
2. If the time window violation P_{tw} of the current solution has not changed for η_{tw} iterations. This criterion helps to diversify if the search gets stuck with a certain (generally small) time window violation and is neither able to find a feasible solution nor to move away due to an ever increasing time window penalty factor that prohibits solutions with a higher time window violation.

To increase the diversification effect of the probabilistic phase, the tabu length ϑ is randomly chosen from a wider interval $[\vartheta_{\text{minProb}}, \vartheta_{\text{maxProb}}]$ during this phase.

4.5.3. Random moves and reset

After TS-DST has not found a solution improving the best overall solution for $\eta_{\text{startRand}}$ iterations, we apply a sequence of φ random moves to the current solution (cf. Tan, Lee, Zhu, & Ou, 2001; Nagata & Bräysy, 2009). Furthermore, if TS-DST has not improved S^* for η_{reset} iterations, the search is restarted from S^*. Due to different tabu list entries and penalties, the search is very likely to take a different path than before.

4.6. Minimization of the number of vehicles

Although inter-route Relocate and 2-opt$*$ moves are able to reduce the number of vehicle routes in a solution, TS-DST is not designed to minimize the number of vehicles. As the literature on VRPTW shows, very specialized additional techniques are often necessary to achieve competitive results concerning the reduction of the number of employed vehicles (see, e.g., Nagata & Bräysy, 2009). An approach commonly applied for heuristic VRPTW methods is to set the number of vehicles m equal to the number of routes of the best known solution reported in the literature (see, e.g., Taillard, Badeau, Gendreau, Guertin, & Potvin, 1997; Cordeau et al., 2001; Ibaraki, Imahori, Kubo, Masuda, Uno, & Yagiura, 2005).

From a practical viewpoint, starting the search with a given number of vehicles can be justified by the fact that in most real-life applications of VRPTW variants, the number of vehicles is given as exogenous parameter determined by the available fleet size and workforce (Ibaraki et al., 2005). If the number of vehicles is not known, the search has to be applied for various values of m. Here, the assumption is made that in practical situations the range of the number of vehicles can be appropriately restricted.

For TS-DST, we distinguish the following cases:

1. In case m is fixed, e.g., due to practical considerations, TS-DST can be adjusted to determine the best possible solution concerning the secondary objectives and employing exactly m vehicles. This is achieved by prohibiting inter-route moves reducing the number of vehicles.
2. If the number of routes of the best known solution m_{best} is available, e.g., from the literature, we proceed as follows. We initially set $m = m_{\text{best}}$, and TS-DST tries to find a feasible solution for η_{feas} iterations. In case a feasible solution is found, the number of vehicles is further reduced until no feasible solution can be found anymore. In contrast, if TS-DST does not succeed, m is increased by one until a feasible solution is found. Up to this point, the search evaluates solutions based on the traveled distance independent of the secondary objective because extensive tests showed that this metric is best suited to find a feasible solution with a restricted number of vehicles. From there, the search continues by optimizing the secondary objective until no improvement has been found for η_{stop} iterations.
3. If m is restricted by an interval $[m_{\min}, m_{\max}]$, for example by some real-life requirement, we apply a binary search approach that uses the so-called fast mode of TS-DST to minimize the number of vehicles. For the number of vehicles given by the search step, the fast mode tries to find a feasible solution for η_{fast} iterations, again evaluating solutions based on the traveled distance. Depending on the success of the run, the next input value is determined by the binary search until a tentative minimum number of vehicles is found. This number is reduced by one and used as input m_{best} of the procedure described for case 2 to potentially further reduce the number of employed vehicles.
4. If no restrictions for m are available like in the case of the newly designed problem instances described in the following section, we determine bounds to constrain the search to an interval for which the binary search described in case 3 can be applied. As lower bound, the number of vehicles defined by capacity restrictions of the VRPTWDST instance is used. To determine the upper bound (UB), this number is multiplied by a factor δ_m until the fast mode of the search finds a feasible solution.

5. Numerical studies

This section describes the numerical experiments to assess the performance of TS-DST and to study the influence of different problem parameters. First, we explain the parameter setting of

Table 1
Overview of the parameter setting of TS-DST chosen for the numerical studies.

Initialization		General TS		Penalties		Probabilistic phase		Diversification	
ϵ_1	0.5	η_{feas}	25,000	α_0, β_0	1, 200	$\eta_{\text{startProb}}$	200	$\eta_{\text{startRand}}$	1000
ϵ_2	0.5	η_{stop}	2500	$\alpha_{\min}, \beta_{\min}$	1	η_{endProb}	50	φ	100
ι	1	η_{fast}	2500	$\alpha_{\max}, \beta_{\max}$	6400	η_{tw}	100	η_{reset}	2500
ν	1	$\vartheta_{\min}$	10	δ	1.2	ξ	250		
α_{init}	1	$\vartheta_{\max}$	25	η_{penalty}	10	$\vartheta_{\text{minProb}}$	100		
β_{init}	5			λ_{div}	2.0	$\vartheta_{\text{maxProb}}$	200		

TS-DST and the experimental environment in Section 5.1. Because no adequate benchmark instances are available for VRPTWDST in the literature, we describe the generation of a comprehensive set of benchmark instances in Section 5.2. Section 5.3 reports the results of TS-DST on the newly generated instances and investigates the influence of different levels of driver familiarity (expressed by different familiarity factors), different geographical distributions of the familiar customers, and different distributions of familiarity among the drivers. To assess the performance of TS-DST, we conduct tests on the Solomon benchmark for the closely related VRPTW and compare the results to the best-performing VRPTW metaheuristics from the literature (Section 5.4). As described in Section 1, benchmark instances and comparison methods proposed for HFVRPTW are not suitable for benchmarking our algorithm: even if vehicle-specific times are included, the benchmarks contain additional features of heterogeneity that are not in the focus of this work.

5.1. Parameter setting and experimental environment

To determine the parameter setting of TS-DST, we carry out a series of pretests on a selection of the generated VRPTWDST benchmark instances. We start from a reasonable parameter setting that was found during the development of the method. Using this setting, we tune one parameter at a time, while keeping all other parameters fixed as proposed in Røpke and Pisinger (2006). We conduct 10 runs of TS-DST with each parameter setting and then choose the setting that produces the best average results. This process is repeated with the next parameter until all parameters are tuned once. This procedure resulted in the setting reported in the overview in Table 1.

The initialization described in Section 4.2 uses parameters $\epsilon_1 = \epsilon_2 = 0.5$ and $\nu = \iota = 1.0$ to evaluate the insertion cost of a customer. For inserting the remaining infeasible customers into the open routes, the generalized cost function with penalty factors $\alpha_{\text{init}} = 1$ and $\beta_{\text{init}} = 5$ is evaluated. TS-DST is run for a maximum of $\eta_{\text{feas}} = 25,000$ iterations to find a feasible solution with a given number of vehicles and continues to minimize the secondary objective until no improvement has been found for $\eta_{\text{stop}} = 2500$ iterations. If the fast mode is used for minimizing the number of vehicles as described in Section 4.6, TS-DST is run for $\eta_{\text{fast}} = 2500$ iterations. The tabu tenure ϑ is drawn from the interval [10, 25].

Concerning the penalty factors of the generalized cost function, we achieved the best results by starting with initial values $\alpha_0 = 1$ and $\beta_0 = 200$ and updating these factors by multiplying or dividing by factor $\delta = 1.2$ every $\eta_{\text{penalty}} = 10$ iterations. The values are restricted to the interval [1, 6400]. For calculating the diversification penalty, we use a factor $\lambda_{\text{div}} = 2.0$.

After $\eta_{\text{tw}} = 100$ iterations without a change in the time window violation of the selected solution and after $\eta_{\text{startProb}} = 200$ iterations without improving the overall best solution, a probabilistic phase is executed for a maximum of $\eta_{\text{endProb}} = 50$ iterations or until a new best overall solution has been found. The list of solutions considered in each iteration of the phase contains the $\xi = 250$ best solutions. During this phase, the tabu tenure is drawn from the interval [100, 200]. If the best overall solution has not improved for $\eta_{startRand} = 1000$ iterations, we perform $\varphi = 100$ random moves to diversify the search. After $\eta_{\text{reset}} = 2500$ iterations without improvement, we reset to the best overall solution.

TS-DST is coded in C++. Studies on the VRPTW instances and a small part of the experiments on the generated VRPTWDST benchmark are performed on a standard desktop computer PC with an Intel Core i7 870 processor at 2.93 gigahertz, 8 gigabytes of RAM, running Windows 7 Professional. In this way, we are able to assess computation times of the proposed method. Due to the high number of generated VRPTWDST instances and the two different objective functions considered, we conduct the remaining tests on a computing cluster. For these tests, run-times are not comparable due to variations in the allocation of resources.

5.2. Generation of VRPTWDST benchmark instances

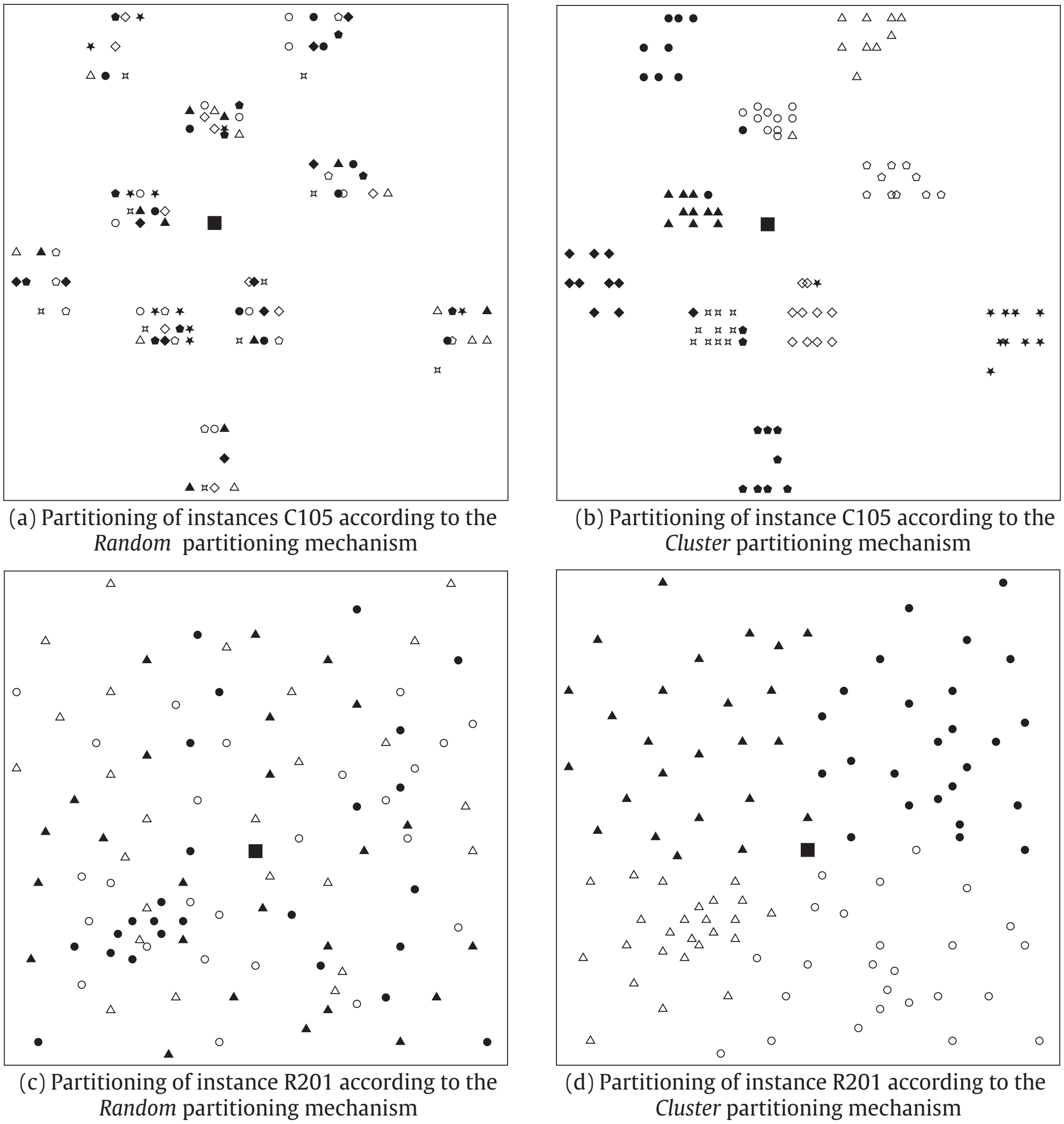
As described in the introduction, no benchmark instances that allow to systematically investigate driver-specific times are available from the literature. We generate VRPTWDST instances based on the well-known 100-customer VRPTW instances of Solomon (1987). The Solomon benchmark consists of 56 instances divided into six instance groups, which are based on different geographical distributions of the customer locations—uniformly random (R), clustered (C), and a combination of clustered and uniformly random (RC)—and different scheduling horizons of the depot $l_0 - e_0$ and capacities of vehicles. Groups C1, R1 and RC1 have a relatively short scheduling horizon and low-capacity vehicles, generally resulting in solutions with a high number of short routes containing only few customers. Groups C2, R2, RC2 have a considerably longer scheduling horizon and high-capacity vehicles, resulting in solutions with a comparatively low vehicle number performing longer routes containing a higher number of customers. Moreover, the instances within a group differ in terms of time window density, i.e., the percentage of customers with a time window (0.25, 0.5, 0.75, 1.0), and time window width, which is defined as the average width of $(l_v - e_v)$ across all customers v in the problem instance.

We define the driver-specific travel and service times of a VRPTWDST instance based on the original distance and service time values of the associated Solomon instance. More precisely, the time t_{vwk} of driver $k \in K$ for traveling arc $(v, w) \in A$ is defined as $t_{vwk} = d_{vw} \cdot (1 - \gamma_{vwk})$, where γ_{vwk} is the familiarity factor of the driver on this arc. In the same manner, the service time s_{vk} for driver k at customer v is computed as $s_{vk} = s_v \cdot (1 - \gamma_{vk})$ with γ_{vk} denoting the familiarity factor of the driver for serving this customer. The value of the familiarity factor is 0 if the driver is unfamiliar with the respective arc or vertex. If the driver is familiar with the arc or vertex, the familiarity factor takes a value in]0.0, 0.5] depending on the degree of familiarity (the higher the value, the higher the familiarity of the driver). This setting is motivated by the assumption of a maximal familiarity gain of 50 percent as explained in Section 2. For a given driver, customers and arcs with a familiarity factor > 0 are referred to as familiar customers and respectively familiar arcs of the driver.

Thus, a VRPTWDST problem instance is implemented by extending the standard VRPTW instance file with the customer familiarity factors $\gamma_{vk}, v \in V$ and the arc familiarity factors $\gamma_{vwk}, v \in V_0, w \in V_{N+1}$ for each driver $k \in K$. The number of available drivers $|K|$ is set to the best-known number of vehicles m_{best} for the associated Solomon instance (e.g., reported in Nagata et al., 2010). Note that this value is an UB on the number of required drivers to solve the VRPTWDST instance because the associated VRPTW instance corresponds to a VRPTWDST instance with only unfamiliar customers and arcs.

In order to study the impact of different familiarity configurations, we generate:

- A base benchmark set in which the sets of familiar customers are disjoint for all drivers (Section 5.2.1).


(a) Partitioning of instances C105 according to the *Random* partitioning mechanism

(b) Partitioning of instance C105 according to the *Cluster* partitioning mechanism

(c) Partitioning of instance R201 according to the *Random* partitioning mechanism

(d) Partitioning of instance R201 according to the *Cluster* partitioning mechanism

Fig. 3. Partitioning of the customer set to obtain the distribution of familiar customers of the drivers using the *Random* and the *Cluster* mechanism on instances with a clustered (C1) and a random (R) geographical customer distribution.

- A second set in which the familiar regions of different drivers overlap in a controlled fashion (Section 5.2.2).
- A third set in which drivers are heterogeneous concerning their familiarity with customers, i.e., there are drivers that are familiar with larger regions while other drivers are completely unfamiliar with the delivery area (Section 5.2.3).

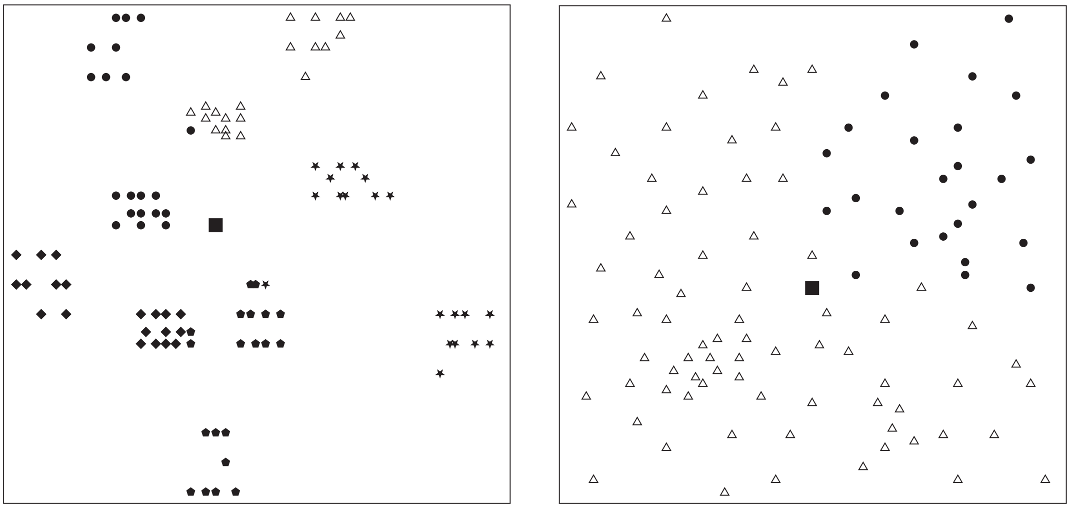
5.2.1. *Disjoint sets of familiar customers*

In this setting, for any customer there is exactly one familiar driver, i.e., the set of customers is partitioned among the drivers. The resulting subsets are chosen to be of approximately equal size, containing $|V|/m_{\text{best}}$ customers. If $|V|$ is not divisible by m_{best}, some arbitrary subsets are chosen to be larger by one. Subset V_k directly defines the familiar customers of driver k. To allow for a systematic study of the impact of the familiarity level of the drivers, we set the familiarity factors of the familiar customers of all drivers to a prespecified value Γ, i.e., $\gamma_{vk} = \Gamma, v \in V_k, k \in K$. We generate test instances with $\Gamma \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$. The familiar arcs of driver k are defined as all arcs starting or ending at a familiar customer. In case of an arc connecting two customers, the familiarity factor of the arc corresponds to the mean value of the familiarity factors of the two customers: $\gamma_{vwk} = \frac{\gamma_{vk}+\gamma_{wk}}{2}$. In case of a depot arc, the familiarity factor of the arc is equal to that of the connected customer.

To determine the subsets V_k, we use two different partitioning mechanisms, called *Random* and *Cluster*, which divide the customer set $V(x)$ into a number of disjoint subsets $V_k(x), k = 1, \dots, m_{\text{best}}(x)$ given a Solomon instance x.

The *Random* mechanism selects the customer subsets in a random fashion, resulting in a geographically dispersed distribution of familiar customers for each driver. Although it is difficult to imagine a practical application with such a distribution, we use this mechanism to generate benchmark instances that are hard to tackle since it is difficult to generate routes that exploit familiarity benefits. Therefore, we expect familiarity gains to be rather low on this type of instances. Fig. 3a and c show the distribution of familiar customers obtained with the *Random* mechanism for a geographically clustered (C105) and a uniformly distributed (R201) Solomon instance. To indicate the familiar customers of each driver, we use different symbols with the following meaning: one driver is familiar with all customers marked by a star, another driver with all those marked by a filled triangle, etc.

The *Cluster* mechanism is inspired by the real-world situation in which familiar customers are likely to appear in clusters that a driver



(a) Instance C105 with overlapping customer sets using $\xi = 2$

(b) Instance R201 with overlapping customer sets using $\xi = 3$

Fig. 4. Examples of instances with overlapping sets of familiar customers.

became familiar with when serving his territory. Therefore, *Cluster* selects the familiar customers of a driver to be geographically close to one another. Consequently, we expect higher familiarity gains if this partitioning mechanism is used because it is easier to generate routes that exploit reduced travel and service times. The *Cluster* mechanism works in a fashion similar to the sweep heuristic of Gillett and Miller (1974): A line centered at the depot is rotated to partition the customers according to their polar angle into subsets of the desired size. The goal of the *Cluster* mechanism is to achieve a partitioning into subsets of familiar customers that matches the geographical distribution of the customers as close as possible, i.e., we want to avoid situations in which the familiar customers of certain drivers are distributed over two or more different geographical clusters. Fig. 3b depicts the results of the *Cluster* mechanism for the Solomon instance C105, which shows a sound matching of the familiarity clusters and the geographical clusters. In Fig. 3d, the result of applying the *Cluster* mechanism on Solomon instance R201 is depicted.

To achieve such a good matching, we restart the sweep procedure $|V|$ times using each of the customers to define the zero degree line with the depot. In this way, $|V|$ different partitionings are generated, whose fit with the geographical distribution is evaluated in the next step: For each of the $|V|$ partitionings, we determine for each subset the central customer, which is defined as the customer with the minimum cumulative distance to all other customers of the subset. The cumulative distance of the central customer to the other customers in the subset is used as an indicator of the subset's compliance with the geographical distribution. Obviously, the measure increases if the familiar customers are distributed over different geographical clusters. Therefore, the partitioning with the minimum cumulative distance over all subsets is finally selected.

For each of the original 56 Solomon instances, we generate VRPTWDST instances with $\Gamma \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ using both partitioning mechanisms (*Random* and *Cluster*). This results in a total of $56 \times 5 \times 2 = 560$ instances in the base benchmark set.

5.2.2. Overlapping sets of familiar customers

The second set of instances is generated to investigate the effect of having a higher number of drivers that are familiar with the same larger customer set. We start from the $56 \times 5 = 280$ instances of the base set that are created with the *Cluster* mechanism, and we generate instances in which $\xi \in \{2, 3, 4\}$ drivers are familiar with the same (larger) customer set. The larger set is composed of the original sets of familiar customers of the ξ drivers. To decide which drivers shall be grouped together, we start from the zero degree line of the sweep that generated the final *Cluster* instance, and we repeatedly merge ξ neighboring subsets in sweep-like fashion as long as we do not exceed the number of available subsets (i.e., number of vehicles) m_{best}. If m_{best} is not dividable by ξ, the $b < \xi$ remaining sets are merged into a larger set and the remaining b drivers are familiar with this last set.

In the described fashion, we generate $3 \times 280 = 840$ instances with overlapping sets of familiar customers. Based on the value of ξ, the instances are grouped into sets *Overlap-2, Overlap-3*, and *Overlap-4*. Fig. 4 shows the distribution of familiar customers as obtained with $\xi = 2$ on Solomon instance C105 and with $\xi = 3$ on Solomon instance R201. In Fig. 4a, two drivers are familiar with each of the customer sets indicated by the same symbol. In Fig. 4b, three drivers are familiar with the customer set indicated by the triangles, the one remaining driver is familiar with the customers indicated by the circles.

5.2.3. Heterogeneous distribution of familiarity among drivers

Finally, we generate instances to study the effect of having drivers that are heterogeneous concerning their familiarity with customers. We generate two types of instances. For the first type, we start from the 280 *Overlap-2* instances. We imagine the extreme case, in which one of the drivers of each two-driver-cluster is familiar with all customers in the cluster, and the other driver is not familiar with any customer. In this way, we generate 280 instances and denote the resulting set as *Heterogeneous-1*.

For the second type, for each Solomon instance and familiarity factor, we generate an instance in which $\lceil m_{\text{best}}/2 \rceil$ of the drivers are familiar with all customers, while the remaining drivers are completely unfamiliar with the entire delivery area. The second group also consists of 280 instances and is denoted as *Heterogeneous-2*.

5.3. Studies on VRPTWDST instances

In this section, we first present and analyze the results of TS-DST on the base VRPTWDST instance set (Section 5.3.1). Here, special emphasis is put on the influence of the level of driver familiarity (expressed by the familiarity factor) and the distribution of familiar customers (Section 5.3.2), and the impact of the geographical distribution of customers (Section 5.3.3). Subsequently, the impact of a higher number of familiar drivers per region is investigated by means of the *Overlap* sets (Section 5.3.4). Finally, the impact of a heterogeneous distribution of familiarity among the drivers is studied on the *Heterogeneous* sets (Section 5.3.5).

Table 2
Aggregate results of TS-DST on the base VRPTWDST benchmark set generated with partitioning mechanisms *Random* and *Cluster* and familiarity factors $\Gamma = \{0.1, 0.2, 0.3, 0.4, 0.5\}$. For each instance group, we report the CNV and the cumulative value of the secondary objective function (CTD, and CWD) based on the best solution found in 10 runs. The deviation to the upper bound corresponding to a pure VRPTW solution (plus an assignment of drivers to routes in case of minimizing working duration) is given in percent as ΔUB. Average run-times of TS-DST are reported in seconds for the case when traveled distance is used as secondary objective.

	Cluster					Random				
	$\Gamma = 0.1$	$\Gamma = 0.2$	$\Gamma = 0.3$	$\Gamma = 0.4$	$\Gamma = 0.5$	$\Gamma = 0.1$	$\Gamma = 0.2$	$\Gamma = 0.3$	$\Gamma = 0.4$	$\Gamma = 0.5$
Number of vehicles										
CNV	397	376	362	354	349	398	387	380	365	360
ΔUB	−1.98	−7.16	−10.62	−12.59	−13.83	−1.73	−4.44	−6.17	−9.88	−11.11
Traveled distance										
CTD	57,027	57,256	57,082	56,265	55,660	57,677	57,700	57,628	58,186	57,500
ΔUB	−1.10	−0.70	−1.00	−2.42	−3.47	0.02	0.07	−0.06	0.91	−0.28
CPU time	64.8	51.2	43.1	34.6	32.6	66.7	59.1	53.0	45.2	39.9
Working duration										
CWD	243,676	231,225	218,403	205,092	190,767	247,883	239,086	228,505	214,662	199,167
ΔUB	−6.49	−10.55	−14.89	−19.45	−24.57	−4.77	−6.91	−10.17	−14.93	−20.46

5.3.1. Performance of TS-DST on the VRPTWDST base benchmark set

We conduct 10 runs of TS-DST on each of the 560 VRPTWDST instances of the base set, using two different objective functions. Both are hierarchical and minimize the number of vehicles first, one minimizes traveled distance (TD) as secondary objective, the other working duration (WD).

Table 2 gives an aggregate view on the obtained results for different instance groups. The instance groups are identified by the partitioning mechanism (*Random, Cluster*) and the familiarity factor ($\Gamma = 0.1, 0.2, 0.3, 0.4, 0.5$). For each instance group, we report the cumulative number of vehicles (CNV) and the cumulative value of the secondary objective (CTD, CWD) based on the best solution found in the 10 TS-DST runs. For the secondary objective TD, we report the average run-times because the experiments were conducted on our desktop PC. Due to the large number of problem instances and test runs, the studies with the secondary objective WD were carried out on the computing cluster and no run-times are reported (the same is true for the studies in the following sections). However, several pretests with smaller problem sets have shown that no systematic change in the run-times is to be expected because of a different objective function. The complete results on an instance basis for all objective functions are reported in Appendix B as comparison values for potential methods addressing the VRPTWDST in the future.

To better assess the obtained solution quality, we use the following UB for the solution of a given VRPTWDST instance *x*. It is determined from the vehicle routes that our TS produces if driver-specific times are ignored, i.e., the familiarity factors of all drivers are set to 0 for all customers. These routes correspond to the VRPTW solution of the original Solomon instance that is associated with the given VRPTWDST instance. We interpret these routes as VRPTWDST solution: If the secondary objective is the minimization of traveled distance, the solution value of this VRPTW solution can be directly used as solution value of the VRPTWDST instance because the assignment of drivers to routes does not influence the traveled distance. However, in case of minimizing working duration, the assignment of drivers to routes affects the objective function value because the routes are evaluated with the driver-specific times of the given VRPTWDST instance. Therefore, we solve an assignment problem that assigns the drivers of the VRPTWDST instance to the constructed routes in the manner that minimizes the working duration. In the following, let S_x^{UB} denote the VRPTWDST solution built from the constructed routes.

The described procedure is performed for all instances of an instance group *G*, and the solution values of the UB are cumulated to $F(\mathrm{UB}) = \sum_{x \in G} F(S_x^{\mathrm{UB}})$, where *F* denotes either the number of vehicles (summing up to CNV), the traveled distance (summing up to CTD) or the working duration (summing up to CWD). Let S_x denote the real VRPTWDST solution of instance *x*, i.e., the solution that TS-DST finds if driver-specific times are considered. Then, the gap of the cumulative objective value $F(\mathrm{VRPTWDST}) = \sum_{x \in G} F(S_x)$ to the described cumulative UB is reported in Table 2 as

$$\Delta\mathrm{UB} = \frac{F(\mathrm{VRPTWDST}) - F(\mathrm{UB})}{F(\mathrm{UB})}.$$

The following sections provide a detailed analysis of the TS-DST results.

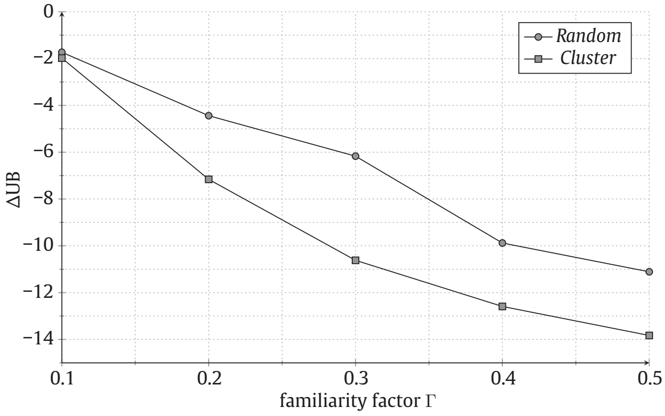


Fig. 5. Influence of different familiarity factors and different distributions of the familiar customers on the number of employed vehicles. As reference point, we use the CNV of the UB solution and plot the gap of the respective instance group.

5.3.2. Influence of familiarity level and distribution of familiar customers

To illustrate the combined influence of different levels of familiarity (expressed by the familiarity factors) and different distributions of the familiar customers (*Cluster* and *Random*) on the number of employed vehicles, Fig. 5 depicts the gap of the CNV values obtained for the different instance groups to the CNV values of the UB solutions.

Higher familiarity factors lead to a decrease of the number of vehicles because the reduced travel and service times allow to generate routes with fewer vehicles that are nevertheless able to fulfill all time window requirements. TS-DST is able to assign the drivers to their familiar customers and thus exploits the reduced times. The number of vehicles also shows consistent behavior concerning the influence of the partitioning mechanism: For each familiarity factor, the number of vehicles in the *Random* generated instance set is higher than in the *Cluster* set. This could be expected, as it is possible to gather more familiarity benefits if the reduced times of a driver occur in a

Table 3
Results of TS-DST on VRPTWDST instances for different instance groups, using traveled distance as secondary objective. For each considered Solomon group, the gap of the TS-DST solution to the UB solution concerning CNV and cumulative traveled distance (CTD) is given.

		Cluster					Random				
		$\Gamma = 0.1$	$\Gamma = 0.2$	$\Gamma = 0.3$	$\Gamma = 0.4$	$\Gamma = 0.5$	$\Gamma = 0.1$	$\Gamma = 0.2$	$\Gamma = 0.3$	$\Gamma = 0.4$	$\Gamma = 0.5$
C1	ΔCNV	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	ΔCTD	0.00	−0.05	−0.05	−0.05	−0.07	0.00	0.00	0.00	−0.02	−0.05
C2	ΔCNV	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	ΔCTD	−0.27	−0.27	−0.28	−0.40	−0.40	−0.14	−0.27	−0.28	−0.39	−0.39
R1	ΔCNV	−2.10	−9.80	−13.29	−17.48	−19.58	−1.40	−4.90	−6.99	−11.89	−13.99
	ΔCTD	−2.12	−2.67	−6.60	−7.91	−8.56	−0.72	−0.40	−1.86	−2.58	−3.09
R2	ΔCNV	0.00	−13.33	−20.00	−20.00	−20.00	−3.33	−10.00	−16.67	−20.00	−20.00
	ΔCTD	−1.54	3.79	5.25	1.93	−0.30	0.32	2.46	4.97	4.19	1.76
RC1	ΔCNV	−3.26	−7.61	−14.13	−14.13	−15.22	−2.17	−5.43	−7.61	−11.96	−14.13
	ΔCTD	−2.72	−6.34	−4.01	−9.92	−11.72	−1.31	−2.99	−3.22	−3.22	−3.36
RC2	ΔCNV	−7.69	−15.38	−19.23	−26.92	−30.77	−7.69	−11.54	−11.54	−23.08	−23.08
	ΔCTD	1.70	3.41	3.26	7.59	6.82	2.64	2.04	0.93	9.24	5.52

clustered region, but it again shows the ability of our algorithm to generate solutions in which drivers are assigned to the right regions and customers. Consequently, the best solution is obtained for a *Cluster* partitioning and a familiarity factor $\Gamma = 0.5$, reducing the CNV by 13.83 percent compared to the UB.

As discussed in Section 1, it is not possible to derive statements about the optimal level of driver familiarity, i.e., the optimal familiarity factor, from these results because we do not consider the cost of driver learning. However, it can be noted that for the *Cluster* partitioning, the gains caused by increasing familiarity factors are not equally distributed: most gains are achieved by increasing the familiarity factor from 0.1 to 0.3, while the increase from 0.3 to 0.5 provides notably less gains. In contrast, for the *Random* partitioning, the gains of higher familiarity factors are more linearly distributed: the gains for increasing the familiarity factor from 0.1 to 0.3 are roughly equal to the gains for increasing from 0.3 to 0.5.

For the secondary objective of minimizing traveled distance, Table 2 shows that no clear trend can be found for the influence of the familiarity factor. The gap sometimes grows for higher familiarity factors and even becomes positive for the *Random* partitioning. These results can be explained by the interdependency of the first and secondary objectives. It is not necessarily possible to obtain a solution with a reduced traveled distance when employing fewer vehicles. The more consistent behavior for the *Cluster* partitioning may be explained as follows: for *Cluster* instances, well-formed, petal-like routes are often sufficient to harvest familiarity benefits in terms of a reduced number of vehicles, thus also leading to reduced traveled distances. For the *Random* instances less compact, zig-zag-like routes are often necessary to reduce the number of vehicles, thus leading to increased traveled distances.

Although minimizing working duration is also only considered as secondary objective, strong familiarity benefits can be observed. This can be explained by the fact that the reduced times directly influence the objective function value. Moreover, the working duration shows entirely consistent behavior: the negative gaps to UB increase (in absolute terms) for higher values of the familiarity factor for both partitioning mechanisms, although the considered solutions employ fewer vehicles than UB. For all considered values of the familiarity factor, the value of ΔUB for the *Cluster* set is clearly below that of the *Random* set because reduced times can better be exploited in the *Cluster* sets. Finally, we observe that the working duration gains caused by higher familiarity factors behave in a roughly linear fashion for *Cluster* and *Random*. Therefore, for both partitioning mechanisms, the “optimal” familiarity factor with regards to working duration is only determined by the preferences of the decision maker (note again that VRPTWDST does not include the cost for achieving higher familiarity factors).

5.3.3. *Influence of geographical distribution of customers and instance design*

To illustrate the influence of the characteristics of the underlying Solomon instances on the obtained VRPTWDST solutions, Table 3 shows the percentage gaps of the CNV and CTD to the respective value of the UB solutions according to the different Solomon groups. For groups C1 and C2, no reduction of the number of vehicles is possible because all instances in these two groups are generated in a way that the number of vehicles is only bounded by the capacity constraint. Time windows in C1 and C2 are positioned around the arrival times at customers obtained by a solution to the problem without time windows. Therefore, reduced times do not allow to generate notably more efficient routes in terms of traveled distance.

A reduction of the number of vehicles is only possible for R and RC instances. Reductions are on average higher for groups R2 and RC2, i.e., the problems with a longer scheduling horizon. One explanation might be that solutions to these instances employ a very low number of vehicles so that reductions lead to high percentage improvements. On the other hand, this characteristic seems to lead to an increase in traveled distance for most cases.

Finally, note that all Solomon groups individually behave in accordance to the above described trend for the influence of familiarity factors and partitioning mechanism on the CNV: the number of vehicles is decreasing with higher familiarity factors for both *Cluster* and *Random*, and the reduction for *Cluster* is always equal or higher than for *Random*.

5.3.4. *Influence of the number of familiar drivers per region*

Next, we consider the scenario in which the familiar regions of drivers overlap, i.e., where several drivers are familiar with the same region that consists of the original familiar regions of the drivers. We conduct 10 runs of TS-DST on each of the instances in the *Overlap* sets and use the best of the 10 runs to determine the CNV and the cumulative secondary objectives CTD and CWD.

Table 4 presents the cumulative solution values for $\xi \in \{2, 3, 4\}$ different familiar drivers per region, i.e., for the sets *Overlap-2*, *Overlap-3*, and *Overlap-4*. The values are given as percentage gaps to those obtained with $\xi = 1$ familiar driver per region, i.e., the associated *Cluster* instance from the base set. Fig. 6 illustrates the behavior for the primary objective of minimizing the number of employed vehicles.

We observe that a higher number of familiar drivers per region allows to reduce the number of employed vehicles compared to the scenario with one driver per region. On average, the higher the number of familiar drivers per region, the stronger the reduction. However, we find that the gain is relatively small even for four familiar drivers per region (< 5 percent). Note that increasing the familiarity factor does not lead to a stronger reduction. Considering that the gaps

Table 4
Effect of different numbers of familiar drivers per region. We report CNV, CTD and CWD for $\xi \in \{2, 3, 4\}$ different drivers per region as gaps to the results obtained with $\xi = 1$ familiar driver per region.

		$\Gamma = 0.1$	$\Gamma = 0.2$	$\Gamma = 0.3$	$\Gamma = 0.4$	$\Gamma = 0.5$	Average
$\xi = 2$	ΔCNV	−2.02	−1.86	−2.21	−1.98	−3.15	−2.24
	ΔCTD	−0.59	−0.32	−1.34	−1.28	−0.59	−0.59
	ΔCWD	−2.78	−4.70	−6.81	−9.84	−10.81	−6.99
$\xi = 3$	ΔCNV	−2.02	−1.86	−2.76	−2.82	−3.72	−2.64
	ΔCTD	0.05	−1.05	−2.04	−1.84	−1.72	−1.32
	ΔCWD	−2.93	−4.96	−7.14	−10.25	−11.7	−7.40
$\xi = 4$	ΔCNV	−3.53	−3.99	−3.04	−3.95	−4.58	−3.82
	ΔCTD	1.06	−0.68	−3.57	−2.45	−3.62	−1.85
	ΔCWD	−3.45	−6.49	−10.45	−13.82	−15.63	−9.99

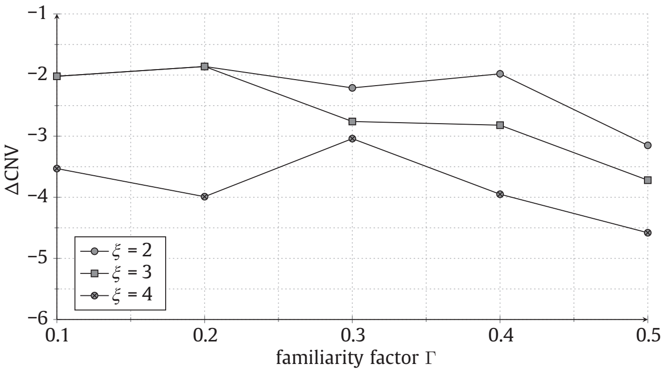


Fig. 6. Influence of different ratios of familiar drivers per region on the number of employed vehicles given as gaps to the values obtained with one familiar driver per region.

are relative to the *Cluster* solution with the same familiarity factor, this result is reasonable.

Concerning the traveled distance, meaningful statements are not possible due to the above described interdependency of this objective and the primary objective of minimizing the number of vehicles. However, for the objective of minimizing working duration, fundamental gains can be observed, which consistently increase with the number of drivers per region and with increasing familiarity factors. Therefore, using several familiar drivers to serve the same (or strongly overlapping) regions seems most recommendable if working duration is in the focus of the decision maker.

5.3.5. *Influence of heterogeneous distribution of driver familiarity*

To study the effect of a heterogenous distribution of driver familiarity, we determine the best of 10 runs of TS-DST for each instance in the sets *Heterogeneous-1* and *Heterogeneous-2*. Table 5 reports the cumulative solution values on the instances where driver familiarity is heterogeneously distributed as gap to the solution values of the corresponding instances with homogeneously distributed familiarity. More precisely, a *Heterogeneous-1* instance with familiarity factor Γ is compared to the respective *Overlap-2* instance with $\Gamma_{\text{base}} = \frac{\Gamma}{2}$. For example, this means that if in a *Heterogeneous-1* instance, one driver is familiar with two clusters with a familiarity factor of $\Gamma = 0.5$ and his counterpart is unfamiliar with both clusters ($\Gamma = 0$), then we compare with the *Overlap-2* instance where both drivers are familiar with both clusters with a familiarity factor of $\Gamma_{\text{base}} = \frac{\Gamma}{2} = 0.25$. The same principle is used for the *Heterogeneous-2* instances. Here, the comparison value is the associated base instance in which all drivers are familiar with all customers with a familiarity factor of $\Gamma_{\text{base}} = \frac{\Gamma}{2}$, where Γ denotes the familiarity factor of the drivers that are familiar with all customers in the *Heterogeneous-2* instance.

Interestingly, we find that heterogeneous distributions of familiarity do not have a negative influence on solution quality compared to homogeneous distributions. On the contrary: for *Heterogeneous-1*, improvements with regard to the working duration can be observed, and for *Heterogeneous-2* considerable improvements in the number of employed vehicles and the working duration can be found, which consistently grow with higher familiarity factors. We conclude that as long as no balancing constraints between driver routes or strict maximal route duration constraints exist, there is no incentive to opt for completely homogeneously distributed driver knowledge. However, given the rather extreme assumptions on the distribution of familiarity in the *Heterogeneous-1* instances and the relatively small resulting improvements, having a heterogeneous distribution should not be considered as a means to increase efficiency. The same holds for the *Heterogeneous-2* instances: although stronger improvements can be observed here, having drivers that are familiar with the entire delivery area will not be a viable option in most practical situations.

5.4. *Performance on standard VRPTW instances*

To be able to assess the solution quality and run-time of TS-DST, we run tests on the Solomon VRPTW instances. In this way, we are able to compare our results to those of the VRPTW methods presented in the literature. TS-DST is run 10 times for each problem instance, starting from the best-known vehicle number reported in the literature for the respective instance as described above.

In Table 6, we present the results of TS-DST, in comparison to the best-performing metaheuristics from the literature. For each method, the solution quality in terms of the number of vehicles and traveled distance is reported. We follow the common procedure and give averages over the problem classes C, R and RC and the CNV and the CTD. For TS-DST, we report the best solution found in the 10 runs. As comparison methods, we use **HY** (Hashimoto & Yagiura, 2008), **LCK** (Le Bouthillier, Crainic, & Kropf, 2005), **BVH** (Bent & Hentenryck, 2004), **PGDR** (Prescott-Gagnon, Desaulniers, & Rousseau, 2009), **HYI** (Hashimoto, Yagiura, & Ibaraki, 2008), **LZ** (Lim & Zhang, 2007), **PR** (Pisinger & Røpke, 2007), **RTI** (Repoussis, Tarantilis, & Ioannou, 2009), **NBD** (Nagata, Bräysy, & Dullaert, 2010), **B** (Bräysy, 2003), and **VCGP** (Vidal, Crainic, Gendreau, & Prins, 2013).

To allow for a fair comparison of the run-times of the different methods, we follow the approach of Gendreau and Tarantilis (2010) and approximately translate the different hardware used in the tests into a comparable time measure. They derive a relative estimated speed compared to a Pentium 4 2.8 gigahertz processor for each of the used machines based on the performance indicators of Dongarra (2014). We use their reported values and estimate our Intel Core i7 processor at 2.93 gigahertz with a speed factor of 1.74 in

Table 5
Effect of a heterogeneous distribution of familiarity among the drivers. We report the cumulative results (based on the best of 10 TS-DST runs) on the *Heterogeneous-1* and *Heterogeneous-2* instance sets as percentage gaps to the results obtained for the corresponding instances with homogeneously distributed familiarity.

	Heterogeneous-1					Heterogeneous-2				
	$\Gamma = 0.1$	$\Gamma = 0.2$	$\Gamma = 0.3$	$\Gamma = 0.4$	$\Gamma = 0.5$	$\Gamma = 0.1$	$\Gamma = 0.2$	$\Gamma = 0.3$	$\Gamma = 0.4$	$\Gamma = 0.5$
Δ_{hom}CNV	0.25	−0.26	−0.82	−0.84	−0.57	−3.26	−6.94	−7.2	−10.3	−11.26
Δ_{hom}CTD	−0.46	−0.10	0.28	0.29	0.12	0.26	−0.13	−3.81	−2.64	−4.29
Δ_{hom}CWD	−0.43	−1.72	−3.61	−5.86	−7.02	−2.06	−5.30	−10.18	−14.83	−17.99

Table 6
The results of TS-DST for VRPTWDST on the Solomon benchmark set in comparison to the best VRPTW methods from the literature. Average results are given for each problem group and best-known solutions are indicated in bold. Moreover, the cumulative number of vehicles (CNV), the cumulative traveled distance (CTD) and run-time related info is provided and the translation into a common time measure is given.

	HY	LCK	BVH	PGDR	HYI	LZ	PR	RTI	NBD	B	VCGP	TS-DST
C1	**10.00**	**10.00**	**10.00**	**10.00**	**10.00**	**10.00**	**10.00**	**10.00**	**10.00**	**10.00**	**10.00**	**10.00**
	828.38	**828.38**	**828.38**	**828.38**	**828.38**	**828.38**	**828.38**	**828.38**	**828.38**	**828.38**	**828.38**	**828.38**
C2	**3.00**	**3.00**	**3.00**	**3.00**	**3.00**	**3.00**	**3.00**	**3.00**	**3.00**	**3.00**	**3.00**	**3.00**
	589.86	**589.86**	**589.86**	**589.86**	**589.86**	**589.86**	**589.86**	**589.86**	**589.86**	**589.86**	**589.86**	**589.86**
R1	11.92	11.92	11.92	**11.92**	11.92	11.92	11.92	11.92	**11.92**	11.92	11.92	11.92
	1216.70	1214.20	1213.25	**1210.34**	1213.18	1213.61	1212.39	1210.82	**1210.34**	1222.12	1210.69	1218.93
R2	2.73	2.73	2.73	2.73	2.73	2.73	2.73	2.73	**2.73**	2.73	2.73	2.73
	961.28	954.32	966.37	955.74	955.61	961.05	957.72	952.67	**951.03**	975.12	951.51	961.74
RC1	11.50	11.50	11.50	**11.50**	11.50	11.50	11.50	11.50	**11.50**	11.50	11.50	11.50
	1384.17	1385.30	1384.22	**1384.16**	1384.25	1385.56	1385.78	1384.30	**1384.16**	1389.58	1384.17	1399.67
RC2	3.25	3.25	3.25	3.25	3.25	3.25	3.25	3.25	**3.25**	3.25	**3.25**	3.25
	1138.37	1129.43	1141.24	1119.44	1120.50	1121.82	1123.49	1119.72	**1119.24**	1128.38	**1119.24**	1135.42
CNV	405	405	405	405	405	405	405	405	**405**	405	405	405
CTD	57,484	57,360	57,567	57,240	57,282	57,368	57,332	57,216	**57,187**	57,710	57,196	57,662
Computer	X2.8G	P850M	SU10	O2.3G	P2.8G	P2.8G	P3G	P3G	O2.4G	P200M	Xe-2.93G	i7/870
Processors	1	5	1	1	1	1	1	1	1	1	1	1
CPU	17.0	12.0	120.0	30.0	16.7	93.2	2.4	17.9	5.0	82.5	2.7	0.4
Runs	1	1	5	5	3	1	10	3	5	1	5	10
Speed	1.26	0.14	0.16	1.41	1.00	1.00	1.07	1.07	1.45	0.03	1.74	1.74
Time	21.48	8.43	94.76	211.85	50.10	93.20	25.77	57.66	36.26	2.19	23.32	6.96

Table 7
Effect of preprocessing rules on the base instance set and the underlying Solomon instance. For each instance group, the average percentage of arcs that can be removed due to these rules is given.

		Cluster					Random				
	VRPTW	$\Gamma = 0.1$	$\Gamma = 0.2$	$\Gamma = 0.3$	$\Gamma = 0.4$	$\Gamma = 0.5$	$\Gamma = 0.1$	$\Gamma = 0.2$	$\Gamma = 0.3$	$\Gamma = 0.4$	$\Gamma = 0.5$
C1	34.89	34.30	33.77	33.26	32.76	32.30	34.33	33.80	33.25	32.65	32.14
C2	30.16	30.02	29.88	29.67	29.38	29.26	30.00	29.84	29.68	29.51	29.38
R1	28.81	27.91	26.96	26.11	25.27	24.42	27.87	26.94	26.05	25.19	24.31
R2	14.70	14.54	14.40	14.24	14.08	13.95	14.52	14.37	14.19	14.03	13.89
RC1	32.96	31.54	30.22	28.89	27.54	26.34	31.49	30.05	28.71	27.35	26.10
RC2	18.53	18.29	18.09	17.90	17.73	17.51	18.27	18.06	17.85	17.64	17.45
Total	26.68	26.10	25.55	25.01	24.46	23.96	26.08	25.51	24.95	24.39	23.88

comparison to their Pentium 4 after running the LINPACK benchmark used in Dongarra (2014) on our machine. Clearly, this technique for comparing the run-times of the different methods is not exact, but it provides a good indication about the speed of the heuristics. To translate into the common time measure, the relative speed factor, the number of employed processors, the number of executed runs and the run-time in minutes are multiplied. All necessary values are reported in Table 6.

TS-DST achieves the best-known CNV of 405 and a CTD of $57,662$. Concerning the traveled distance, TS-DST ranks 11th among the 12 best performing VRPTW heuristics with only small differences to the CTD found by the best methods: The gap to the CTD of Nagata et al. (2010) is 0.8 percent. This seems a convincing result because TS-DST was designed as solution method for VRPTWDST and the (rather basic) parameter tuning was carried out on VRPTWDST instances. TS-DST is the only TS able to solve the Solomon VRPTW benchmark with a CNV of 405 vehicles and thus the best-performing TS for the VRPTW proposed in the literature. The formerly best-performing TS heuristics for VRPTW are Taillard et al. (1997) with a CNV of 410 and a CTD of $57,523$ and Cordeau et al. (2001) with a CNV of 407 and a CTD of $57,556$.

Concerning run-time, our approach needs on average 24 seconds to solve each instance. This translates to a value of 6.96 for the common time measure and thus, TS-DST is the second fastest of the methods presented in the table. Run-times to the formerly best TSs are not comparable as Taillard et al. (1997) and Cordeau et al. (2001) do not report the necessary values in their papers. In comparison to the results of the VRPTWDST test sets, a reduction of the run-times for VRPTW can be noticed. This can be explained by the additional calculations necessary to apply the time travel approach for time window handling in a driver-specific manner.

6. Summary and Conclusions

This paper investigates the VRPTWDST, a routing problem that integrates different extents of available driver knowledge by means of driver-specific times. We present TS-DST as metaheuristic solution method. We generate a comprehensive set of VRPTWDST benchmark instances covering different familiarity levels and customer arrangements in order to carry out a systematic investigation of the problem. We find that consideration of driver knowledge in the route planning clearly increases the efficiency of vehicle routes, an effect that intensifies for higher familiarity levels. Moreover, increased familiarity benefits are produced if the customers, that drivers are familiar with, are geographically contiguous. A higher number of drivers that is familiar with the same (larger) region provides increased benefits compared to a scenario where each driver is only familiar with a dedicated (smaller) region.

TS-DST is able to prove its performance on Solomon test instances of the closely related VRPTW, yielding high-quality solutions in short time. By accomplishing a CNV of 405 vehicles, TS-DST is the most successful TS method proposed for the VRPTW. The presented TS-DST shows how a simple metaheuristic method can be enhanced with diversification methods to accomplish a very good solution quality for the intensively studied VRPTW, nearly competing with sophisticated approaches. TS-DST can consequently provide well-founded decision support for any logistics company whose routing operations have to take driver knowledge effects into account.

Table 8
Results of TS-DST for the secondary objective of minimizing traveled distance on the VRPTWDST benchmark sets generated with the *Random* mechanism. The best solution found in 10 runs is reported in columns *m* and TD and the average run-time in seconds in column CPU.

	$\Gamma = 0.1$			$\Gamma = 0.2$			$\Gamma = 0.3$			$\Gamma = 0.4$			$\Gamma = 0.5$		
	m	TD	CPU	*m*	TD	CPU	*m*	TD	CPU	*m*	TD	CPU	*m*	TD	CPU
c101	10	828.94	4.63	10	828.94	4.69	10	828.94	4.75	10	828.94	4.80	10	828.94	4.86
c102	10	828.94	6.41	10	828.94	6.45	10	828.94	6.54	10	828.94	6.54	10	828.94	6.65
c103	10	828.07	7.58	10	828.07	7.84	10	828.07	8.23	10	828.07	8.32	10	828.07	8.92
c104	10	824.78	12.80	10	824.78	14.09	10	824.78	11.85	10	822.79	10.82	10	822.79	13.49
c105	10	828.94	5.07	10	828.94	5.11	10	828.94	5.20	10	828.94	5.31	10	828.94	5.35
c106	10	828.94	5.23	10	828.94	5.28	10	828.94	5.37	10	828.94	5.44	10	828.94	5.51
c107	10	828.94	5.42	10	828.94	5.48	10	828.94	5.54	10	828.94	5.65	10	828.94	5.72
c108	10	828.94	5.96	10	828.94	5.92	10	828.94	6.03	10	828.94	6.08	10	828.94	6.21
c109	10	828.94	6.69	10	828.94	6.72	10	828.94	6.80	10	828.94	6.84	10	827.38	8.97
Average C1	10	828.38	6.64	10	828.38	6.84	10	828.38	6.70	10	828.16	6.64	10	827.98	7.30
c201	3	591.56	7.17	3	591.56	7.20	3	591.56	7.30	3	588.88	7.20	3	588.88	7.23
c202	3	591.56	10.18	3	591.56	10.15	3	591.56	10.11	3	588.88	14.89	3	588.88	10.15
c203	3	588.49	14.23	3	585.27	14.45	3	585.27	11.96	3	585.27	14.97	3	585.27	15.57
c204	3	587.71	17.77	3	584.49	17.76	3	584.49	21.49	3	584.49	18.89	3	584.49	18.69
c205	3	588.49	7.18	3	588.49	7.04	3	588.49	7.19	3	588.49	7.21	3	588.49	7.44
c206	3	588.49	8.08	3	588.49	7.99	3	588.49	8.71	3	588.49	8.81	3	588.49	7.86
c207	3	588.29	8.80	3	588.29	8.84	3	588.29	8.81	3	588.29	9.04	3	588.29	8.38
c208	3	588.32	8.20	3	588.32	9.70	3	588.29	8.27	3	588.29	8.52	3	588.29	8.60
Average C2	3.00	589.11	10.20	3.00	588.31	10.39	3.00	588.30	10.48	3.00	587.63	11.19	3.00	587.63	10.49
r101	18	1606.07	194.90	17	1674.16	195.00	17	1611.47	184.53	16	1622.30	185.75	16	1558.53	177.86
r102	17	1447.36	264.09	15	1440.30	213.71	15	1433.02	215.91	14	1481.84	215.00	14	1376.89	194.16
r103	13	1257.49	204.15	12	1369.66	194.54	12	1303.54	190.17	11	1195.22	125.77	11	1167.72	126.44
r104	9	1007.39	60.38	9	987.19	61.09	9	969.21	57.81	9	947.49	59.53	8	1024.01	15.73
r105	13	1462.39	128.28	13	1410.12	124.81	13	1381.95	127.05	12	1419.59	115.35	12	1375.88	110.78
r106	12	1263.29	136.73	11	1314.50	113.56	11	1241.05	108.30	10	1265.66	85.32	10	1219.85	80.97
r107	10	1080.89	90.39	10	1072.09	93.19	9	1108.48	65.33	9	1059.46	54.49	8	1197.18	42.80
r108	9	957.04	57.03	9	941.39	57.80	9	925.20	59.99	8	975.64	17.31	8	944.22	13.77
r109	11	1205.27	100.87	11	1174.45	104.07	10	1236.17	97.31	10	1202.77	79.22	10	1171.51	80.11
r110	10	1128.61	88.34	10	1107.08	92.46	10	1078.05	92.08	9	1079.99	54.51	9	1061.56	55.41
r111	10	1102.07	89.69	10	1086.05	90.02	9	1097.96	70.12	9	1050.56	55.95	9	1029.39	55.50
r112	9	1003.76	63.56	9	991.06	61.68	9	968.97	64.76	9	949.38	64.14	8	1048.90	10.35
Average R1	10.33	1134.52	90.58	10.22	1120.44	88.74	9.89	1111.89	82.53	9.44	1105.62	65.09	9.11	1119.17	51.71
r201	4	1248.49	99.13	3	1439.76	61.08	3	1440.03	60.38	3	1385.21	60.81	3	1346.87	59.69
r202	3	1177.11	84.36	3	1159.18	77.50	3	1129.93	83.00	3	1123.35	77.24	3	1099.87	80.93
r203	3	939.54	97.35	3	935.13	99.70	2	1149.74	19.62	2	1079.65	30.33	2	1071.39	24.63
r204	2	822.66	35.21	2	819.13	28.17	2	797.46	31.21	2	784.58	30.01	2	763.15	31.41
r205	3	1005.05	71.12	3	994.59	78.71	3	1000.63	78.56	2	1208.19	15.37	2	1136.55	13.41
r206	2	1076.74	85.19	2	1024.90	19.70	2	989.35	22.18	2	973.44	23.29	2	958.93	22.20
r207	2	883.50	23.55	2	860.67	29.17	2	857.32	29.27	2	853.70	30.76	2	844.66	24.54
r208	2	730.62	28.01	2	733.07	29.88	2	726.29	30.94	2	722.19	23.88	2	719.55	34.93
r209	3	915.07	84.78	3	887.77	89.28	2	1098.65	26.88	2	1019.92	18.01	2	983.49	25.12
r210	3	949.52	89.30	2	1139.36	87.66	2	1089.68	28.22	2	1072.67	29.79	2	1055.61	24.28
r211	2	864.83	21.74	2	845.81	31.40	2	826.30	18.49	2	799.14	23.65	2	785.00	30.13
Average R2	2.44	909.73	59.58	2.33	915.60	54.85	2.11	948.38	31.71	2.00	945.94	25.01	2.00	924.26	25.63
rc101	14	1685.39	107.07	14	1629.88	106.54	13	1644.96	94.38	13	1563.74	93.48	12	1633.25	85.79
rc102	12	1503.25	97.19	11	1547.76	77.80	11	1468.57	70.13	10	1523.71	66.71	10	1424.50	46.52
rc103	10	1305.20	55.47	10	1246.04	48.88	10	1222.23	48.70	9	1294.62	17.40	9	1283.44	11.23
rc104	10	1118.42	57.50	9	1170.48	24.75	9	1169.87	14.63	9	1110.60	18.39	9	1123.56	15.24
rc105	13	1626.49	126.67	12	1544.03	95.68	12	1486.86	91.53	11	1503.36	70.22	11	1453.05	67.06
rc106	11	1366.86	65.38	11	1340.55	66.35	11	1334.18	65.12	10	1417.68	65.20	10	1367.78	42.62
rc107	10	1312.23	75.26	10	1268.77	46.54	10	1275.88	47.79	10	1232.28	48.27	9	1363.72	18.69
rc108	10	1132.60	53.49	10	1114.57	53.18	9	1234.71	34.63	9	1190.92	15.21	9	1172.26	13.86
Average RC1	11.25	1381.31	79.76	10.88	1357.76	64.96	10.63	1354.66	58.37	10.13	1354.61	49.36	9.88	1352.70	37.63
rc201	4	1394.81	98.68	4	1356.99	99.87	4	1358.77	97.88	3	1651.83	62.94	3	1593.78	65.53
rc202	3	1326.40	82.46	3	1291.46	82.63	3	1277.85	80.09	3	1270.55	83.47	3	1257.01	83.32
rc203	3	1066.66	96.18	3	1030.14	93.59	3	1015.40	92.29	2	1251.35	51.26	2	1182.23	52.91
rc204	2	945.44	100.40	2	931.74	47.97	2	903.04	50.72	2	888.86	42.99	2	859.43	27.24
rc205	3	1566.16	112.17	3	1484.43	78.39	3	1464.99	84.24	3	1464.85	80.13	3	1420.21	77.40
rc206	3	1140.98	73.67	3	1117.61	76.13	3	1116.14	76.49	3	1111.19	72.90	3	1103.49	74.45
rc207	3	1055.42	90.56	3	1053.72	92.38	3	1048.91	88.15	2	1358.43	90.72	2	1255.91	30.96
rc208	3	827.58	106.13	2	1002.38	37.82	2	982.97	25.83	2	925.21	25.35	2	912.75	34.78
Average RC2	3.00	1165.43	95.03	2.88	1158.56	76.10	2.88	1146.01	74.46	2.50	1240.28	63.72	2.50	1198.10	55.82
CNV	398			387			380			365			360		
CTD		57677			57700			57628			58186			57500	
Average CPU			66.7			59.1			53.0			45.2			39.9

Table 9
Results of TS-DST for the secondary objective of minimizing traveled distance on the VRPTWDST benchmark sets generated with the *Cluster* mechanism. The best solution found in 10 runs is reported in columns m and TD and the average run-time in seconds in column CPU.

	$\Gamma = 0.1$			$\Gamma = 0.2$			$\Gamma = 0.3$			$\Gamma = 0.4$			$\Gamma = 0.5$		
	m	TD	CPU	m	TD	CPU	m	TD	CPU	m	TD	CPU	m	TD	CPU
c101	10	828.94	4.66	10	828.94	4.73	10	828.94	4.81	10	828.94	4.82	10	828.94	4.86
c102	10	828.94	6.36	10	828.94	6.36	10	828.94	6.38	10	828.94	6.46	10	828.94	6.52
c103	10	828.07	7.54	10	828.07	7.55	10	828.07	7.74	10	828.07	7.87	10	828.07	8.07
c104	10	824.78	12.05	10	822.79	13.10	10	822.79	13.20	10	822.79	14.90	10	822.79	15.31
c105	10	828.94	5.14	10	828.94	5.15	10	828.94	5.25	10	828.94	5.28	10	828.94	5.33
c106	10	828.94	5.25	10	828.94	5.33	10	828.94	5.37	10	828.94	5.48	10	828.94	5.50
c107	10	828.94	5.47	10	828.94	5.43	10	828.94	5.50	10	828.94	5.62	10	828.94	5.66
c108	10	828.94	5.96	10	828.94	5.96	10	828.94	6.07	10	828.94	6.23	10	828.94	6.57
c109	10	828.94	6.70	10	827.38	7.62	10	827.38	6.85	10	827.26	6.93	10	825.65	6.90
Average C1	10.00	828.38	6.57	10.00	827.98	6.80	10.00	827.98	6.80	10.00	827.97	7.07	10.00	827.79	7.19
c201	3	591.56	7.26	3	591.56	7.34	3	591.56	7.10	3	588.88	8.92	3	588.88	8.17
c202	3	591.56	9.89	3	591.56	9.97	3	591.56	9.75	3	588.88	9.85	3	588.88	9.88
c203	3	585.27	11.46	3	585.27	14.24	3	585.27	13.44	3	585.27	13.43	3	585.27	14.23
c204	3	584.49	17.77	3	584.49	17.42	3	584.49	20.10	3	584.49	18.93	3	584.49	17.09
c205	3	588.49	7.05	3	588.49	7.23	3	588.49	7.12	3	588.49	7.30	3	588.49	7.40
c206	3	588.49	8.54	3	588.49	7.98	3	588.49	7.74	3	588.49	8.67	3	588.49	8.97
c207	3	588.29	9.04	3	588.29	8.89	3	588.29	9.38	3	587.89	10.12	3	587.89	10.44
c208	3	588.32	9.50	3	588.32	9.39	3	588.29	9.19	3	588.29	10.30	3	588.29	10.51
Average C2	3	588.31	10.07	3	588.31	10.31	3	588.30	10.48	3	587.58	10.94	3	587.58	10.84
r101	18	1608.59	194.30	17	1619.91	191.65	16	1551.56	164.44	16	1536.14	174.12	15	1574.50	159.99
r102	17	1444.67	265.10	15	1457.19	218.57	15	1379.01	215.86	14	1372.01	192.00	14	1358.34	192.50
r103	13	1261.37	204.43	12	1276.87	183.30	11	1198.12	126.63	10	1152.75	90.50	10	1121.88	86.86
r104	9	986.64	62.41	9	942.61	54.38	8	939.98	14.79	8	910.68	15.98	8	896.12	14.96
r105	13	1423.89	122.35	12	1413.15	122.24	12	1330.78	105.20	11	1349.07	83.56	10	1425.62	79.74
r106	11	1314.91	119.31	11	1221.56	111.29	10	1211.56	87.39	10	1193.47	78.12	9	1245.11	49.33
r107	10	1048.99	89.25	9	1053.08	57.76	8	1082.37	16.78	8	992.83	13.20	8	960.26	15.62
r108	9	932.50	58.75	8	931.37	17.01	8	892.50	19.12	8	870.65	20.11	8	861.62	18.21
r109	11	1174.15	99.54	10	1176.91	73.62	10	1131.55	75.81	9	1135.71	45.63	9	1091.13	43.80
r110	10	1094.26	91.50	9	1104.17	55.72	9	1028.17	55.52	8	1085.01	14.08	8	1010.48	12.95
r111	10	1066.32	88.56	9	1043.59	52.92	9	995.65	53.99	8	992.06	17.48	8	963.58	20.80
r112	9	961.33	61.95	8	996.67	52.69	8	920.35	14.04	8	879.14	18.08	8	865.77	17.66
Average R1	10.22	1111.44	88.18	9.44	1098.12	66.40	9.11	1059.21	49.18	8.67	1045.40	34.03	8.44	1035.52	30.34
r201	4	1245.18	97.90	3	1438.73	60.61	3	1378.45	63.39	3	1343.04	59.28	3	1298.62	61.10
r202	3	1164.86	79.70	3	1113.21	79.58	3	1105.96	80.14	3	1087.14	78.71	3	1079.28	80.95
r203	3	937.92	94.41	3	927.67	98.00	2	1066.05	33.72	2	1057.21	36.21	2	1026.46	31.08
r204	2	813.48	36.41	2	804.88	26.28	2	788.33	34.06	2	773.13	24.54	2	764.34	24.18
r205	3	1001.71	75.41	3	1008.61	73.65	2	1282.59	74.10	2	1150.60	21.80	2	1115.55	20.91
r206	3	909.77	91.13	2	1043.82	26.15	2	1001.94	20.24	2	980.76	20.82	2	974.91	18.66
r207	2	883.52	39.00	2	872.31	30.89	2	845.81	26.30	2	838.25	26.46	2	832.14	21.11
r208	2	728.69	29.88	2	730.06	32.76	2	726.28	28.06	2	720.80	20.33	2	726.15	27.27
r209	3	906.16	86.25	2	1099.87	54.28	2	1056.46	32.88	2	1008.83	20.65	2	960.25	23.43
r210	3	949.13	90.80	2	1105.23	36.43	2	1071.27	28.89	2	1020.64	28.53	2	992.73	27.94
r211	2	875.84	21.17	2	835.93	23.96	2	811.47	29.25	2	803.24	27.11	2	777.04	26.98
Average R2	2.56	889.58	62.72	2.22	936.49	44.71	2.00	961.13	34.17	2.00	928.16	25.16	2.00	907.73	24.62
rc101	14	1624.34	104.91	13	1589.19	90.16	12	1610.57	88.41	12	1528.71	72.94	12	1463.10	76.64
rc102	12	1446.74	97.96	11	1437.23	68.24	10	1533.98	60.02	10	1374.31	42.85	10	1363.84	40.92
rc103	10	1276.52	49.34	10	1224.91	47.44	9	1320.31	24.84	9	1243.41	15.55	9	1196.85	11.78
rc104	9	1209.38	44.40	9	1107.53	17.15	9	1089.31	19.07	9	1043.06	18.61	9	1033.30	16.70
rc105	13	1609.55	128.82	12	1461.09	88.51	11	1530.70	79.05	11	1420.35	66.97	11	1367.23	67.09
rc106	11	1338.16	61.84	11	1295.07	64.12	10	1287.25	37.26	10	1203.80	40.28	9	1285.41	9.80
rc107	10	1276.71	50.34	10	1227.21	46.23	9	1307.75	42.52	9	1235.80	9.39	9	1152.32	15.53
rc108	10	1111.59	54.24	9	1145.40	13.22	9	1068.44	16.46	9	1037.09	14.17	9	1022.49	14.60
Average RC1	11.13	1361.62	73.98	10.63	1310.95	54.39	9.88	1343.54	45.95	9.88	1260.82	35.09	9.75	1235.57	31.63
rc201	4	1378.03	98.67	3	1589.24	62.72	3	1566.90	60.02	3	1522.07	59.18	3	1478.19	60.02
rc202	3	1325.29	83.44	3	1289.34	84.11	3	1272.29	85.83	3	1245.64	84.22	2	1713.57	77.48
rc203	3	1040.73	95.18	3	1001.54	97.48	2	1188.11	67.43	2	1222.14	21.84	2	1113.52	30.48
rc204	2	940.05	65.68	2	921.70	27.48	2	870.31	24.66	2	843.49	31.48	2	821.70	37.13
rc205	3	1558.69	78.15	3	1482.37	76.86	3	1464.75	73.48	3	1410.72	72.70	3	1366.92	73.86
rc206	3	1134.95	75.57	3	1117.64	76.76	3	1109.14	71.02	2	1391.79	56.42	2	1270.44	18.86
rc207	3	1052.83	93.74	3	1030.11	92.93	3	1022.41	89.40	2	1256.39	21.79	2	1125.97	22.08
rc208	3	807.46	104.98	2	961.07	36.79	2	885.53	26.82	2	881.00	28.62	2	812.26	27.30
Average RC2	3.00	1154.75	86.93	2.75	1174.13	69.39	2.63	1172.43	62.34	2.38	1221.66	47.03	2.25	1212.82	43.40
CNV	397			376			362			354			349		
CTD		57027			57256			57082			56265			55659	
Average CPU			64.8			51.2			43.1			34.6			32.6

Table 10
Results of TS-DST for the secondary objective of minimizing working duration on the VRPTWDST benchmark sets generated with the *Random* mechanism. The best solution found in 10 runs is reported in columns *m* and WD.

	$\Gamma = 0.1$		$\Gamma = 0.2$		$\Gamma = 0.3$		$\Gamma = 0.4$		$\Gamma = 0.5$	
	m	WD	*m*	WD	*m*	WD	*m*	WD	*m*	WD
c101	10	9583.95	10	9434.01	10	9255.85	10	9057.90	10	8645.71
c102	10	9571.76	10	9368.29	10	9090.35	10	8332.91	10	7731.29
c103	10	9566.64	10	9275.52	10	8713.50	10	7889.27	10	7156.01
c104	10	9544.89	10	9061.47	10	8306.39	10	7309.32	10	6362.21
c105	10	9560.17	10	9312.68	10	9026.51	10	8631.05	10	7968.11
c106	10	9563.77	10	9348.15	10	8968.86	10	8557.57	10	7955.27
c107	10	9559.69	10	9290.51	10	8849.37	10	8322.11	10	7687.62
c108	10	9559.69	10	9251.35	10	8718.00	10	8013.51	10	7426.01
c109	10	9552.03	10	9183.30	10	8464.41	10	7716.09	10	6990.04
Average C1	10.00	9562.51	10.00	9280.59	10.00	8821.47	10.00	8203.30	10.00	7546.92
c201	3	9331.71	3	9295.09	3	9172.76	3	8975.55	3	8792.56
c202	3	9248.77	3	9031.62	3	8942.25	3	8663.72	3	7652.06
c203	3	9231.73	3	8836.30	3	8075.28	3	7474.47	3	7028.77
c204	3	9166.18	3	8479.90	3	7699.91	3	6967.83	3	6265.37
c205	3	9216.11	3	9032.01	3	8858.45	3	8636.26	3	8059.02
c206	3	9213.06	3	8905.06	3	8600.91	3	8148.83	3	7693.07
c207	3	9215.72	3	8773.49	3	8306.10	3	7851.54	3	7376.12
c208	3	9213.06	3	8711.12	3	8294.34	3	7758.69	3	7255.76
Average C2	3.00	9229.54	3.00	8883.07	3.00	8493.75	3.00	8059.61	3.00	7515.34
r101	18	3234.05	17	3128.37	17	3030.77	16	2902.28	16	2817.01
r102	17	2893.27	15	2654.13	15	2585.16	14	2520.96	14	2414.43
r103	13	2421.84	12	2406.04	12	2293.43	11	2065.98	11	1997.12
r104	9	1990.74	9	1931.02	9	1854.18	9	1776.55	8	1668.82
r105	13	2516.37	13	2430.73	13	2353.31	12	2262.50	12	2164.98
r106	12	2265.89	11	2218.62	11	2108.62	10	1994.61	10	1872.53
r107	10	2062.66	10	1976.99	9	1934.55	9	1821.57	8	1708.50
r108	9	1920.47	9	1843.42	9	1750.91	8	1698.45	8	1572.55
r109	11	2203.86	11	2105.56	10	2087.31	10	1987.03	10	1826.13
r110	10	2119.15	10	2035.77	10	1925.06	9	1845.38	9	1685.03
r111	10	2086.92	10	2024.88	9	1955.01	9	1842.36	9	1740.69
r112	9	1973.05	9	1893.93	9	1794.08	9	1684.89	8	1635.75
Average R1	10.33	2126.57	10.22	2051.21	9.89	1973.67	9.44	1879.26	9.11	1763.89
r201	4	2867.68	3	2505.94	3	2454.66	3	2318.69	3	2219.81
r202	3	2327.48	3	2282.67	3	2207.00	3	2167.85	3	2057.22
r203	3	2126.71	3	2053.93	2	1900.86	2	1753.98	2	1662.85
r204	2	1768.52	2	1647.68	2	1468.46	2	1360.18	2	1307.39
r205	3	2143.95	3	2042.30	3	1957.98	2	1733.73	2	1645.77
r206	2	1994.20	2	1873.58	2	1756.25	2	1643.91	2	1575.01
r207	2	1793.92	2	1663.53	2	1511.52	2	1376.51	2	1238.90
r208	2	1652.26	2	1521.14	2	1369.60	2	1246.14	2	1127.04
r209	3	2074.44	3	1955.27	2	1757.75	2	1603.94	2	1477.99
r210	3	2062.00	2	1982.48	2	1853.72	2	1740.60	2	1620.19
r211	2	1778.66	2	1609.56	2	1483.74	2	1344.93	2	1224.80
Average R2	2.44	1932.74	2.33	1816.61	2.11	1673.32	2.00	1533.77	2.00	1431.10
rc101	14	2818.91	14	2718.95	13	2595.45	13	2474.42	12	2361.42
rc102	12	2539.56	11	2423.54	11	2278.13	10	2206.00	10	2051.62
rc103	10	2249.37	10	2132.13	10	2019.24	9	1957.34	9	1833.17
rc104	10	2101.67	9	2052.74	9	1965.48	9	1829.04	9	1730.49
rc105	13	2743.21	12	2566.28	12	2409.91	11	2324.82	11	2232.29
rc106	11	2368.19	11	2286.66	11	2209.45	10	2132.24	10	1979.08
rc107	10	2279.65	10	2184.03	10	2094.72	10	1998.14	9	1926.65
rc108	10	2131.41	10	2025.99	9	2023.63	9	1927.52	9	1817.93
Average RC1	11.25	2404.00	10.88	2298.79	10.63	2199.50	10.13	2106.19	9.88	1991.58
rc201	4	2967.93	4	2905.14	4	2769.68	3	2448.79	3	2262.17
rc202	3	2509.21	3	2436.80	3	2309.28	3	2113.25	3	2021.88
rc203	3	2139.98	3	2057.74	3	1990.58	2	1820.32	2	1718.18
rc204	2	1889.04	2	1806.79	2	1690.03	2	1587.97	2	1450.97
rc205	3	2608.90	3	2525.24	3	2414.75	3	2278.16	3	2195.91
rc206	3	2357.10	3	2254.69	3	2092.30	3	1950.77	3	1794.25
rc207	3	2149.51	3	2099.02	3	1971.85	2	1858.48	2	1648.67
rc208	3	1912.47	2	1844.60	2	1713.40	2	1591.40	2	1438.67
Average RC2	3.00	2316.77	2.88	2241.25	2.88	2118.98	2.50	1956.14	2.50	1816.34
CNV	398		387		380		365		360	
CNV		247943		238698		227285		213498		198769

Table 11
Results of TS-DST for the secondary objective of minimizing working duration on the VRPTWDST benchmark sets generated with the *Cluster* mechanism. The best solution found in 10 runs is reported in columns *m* and WD.

	$\Gamma = 0.1$		$\Gamma = 0.2$		$\Gamma = 0.3$		$\Gamma = 0.4$		$\Gamma = 0.5$	
	m	WD	*m*	WD	*m*	WD	*m*	WD	*m*	WD
c101	10	9381.05	10	9230.48	10	9063.84	10	8926.60	10	8763.62
c102	10	9307.14	10	8563.07	10	8110.61	10	7746.53	10	7406.51
c103	10	9228.87	10	8348.18	10	7692.83	10	7103.38	10	6696.33
c104	10	9039.03	10	8090.96	10	7256.42	10	6511.20	10	5761.96
c105	10	9108.74	10	8857.08	10	8449.57	10	8056.21	10	7749.02
c106	10	9116.34	10	8732.08	10	8356.57	10	7943.84	10	7555.84
c107	10	8934.28	10	8588.13	10	8030.66	10	7682.15	10	7402.21
c108	10	8932.56	10	8374.42	10	7873.64	10	7460.00	10	7070.58
c109	10	8915.26	10	8040.37	10	7506.81	10	6985.62	10	6638.63
Average C1	10.00	9107.03	10.00	8536.09	10.00	8037.88	10.00	7601.73	10.00	7227.19
c201	3	9307.61	3	9259.88	3	9153.64	3	9015.96	3	8861.14
c202	3	9070.63	3	8996.20	3	8913.51	3	8828.72	3	7777.11
c203	3	8893.12	3	8174.89	3	7602.98	3	7180.87	3	6954.49
c204	3	8751.38	3	8037.65	3	7425.86	3	6662.80	3	6034.82
c205	3	9058.30	3	8961.80	3	8802.10	3	8581.46	3	8073.32
c206	3	8905.27	3	8688.19	3	8402.86	3	7892.50	3	7375.14
c207	3	8688.28	3	8284.65	3	8018.19	3	7539.07	3	7168.51
c208	3	8661.30	3	8378.12	3	7961.11	3	7486.05	3	7101.40
Average C2	3.00	8916.99	3.00	8597.67	3.00	8285.03	3.00	7898.43	3.00	7418.24
r101	18	3229.63	17	3077.07	16	2875.61	16	2767.17	15	2650.92
r102	17	2892.76	15	2679.43	15	2586.89	14	2415.69	14	2348.05
r103	13	2374.91	12	2256.92	11	2031.71	10	1873.84	10	1787.28
r104	9	1894.66	9	1726.85	8	1612.20	8	1464.58	8	1336.43
r105	13	2453.98	12	2328.62	12	2163.87	11	2045.65	10	1976.11
r106	11	2236.72	11	2022.85	10	1955.31	10	1804.31	9	1748.00
r107	10	1995.88	9	1825.32	8	1704.71	8	1536.18	8	1424.78
r108	9	1790.73	8	1675.49	8	1515.02	8	1367.68	8	1208.96
r109	11	2133.12	10	2002.14	10	1828.33	9	1702.55	9	1566.24
r110	10	2038.90	9	1889.70	9	1672.35	8	1597.27	8	1426.74
r111	10	1982.24	9	1850.83	9	1663.14	8	1566.11	8	1437.37
r112	9	1837.54	8	1756.15	8	1514.03	8	1345.42	8	1189.50
Average R1	10.22	2040.42	9.44	1897.55	9.11	1736.55	8.67	1603.31	8.44	1479.35
r201	4	2846.18	3	2501.27	3	2437.68	3	2317.93	3	2232.12
r202	3	2310.11	3	2240.59	3	2181.59	3	2083.36	3	2004.01
r203	3	2096.68	3	2043.29	2	1805.03	2	1716.38	2	1651.52
r204	2	1725.81	2	1551.64	2	1413.14	2	1344.75	2	1323.20
r205	3	2105.45	3	1954.35	2	1872.79	2	1676.38	2	1608.68
r206	3	1930.02	2	1857.80	2	1705.49	2	1595.90	2	1555.77
r207	2	1746.57	2	1600.15	2	1560.57	2	1293.91	2	1188.48
r208	2	1593.66	2	1460.46	2	1315.45	2	1198.77	2	1125.80
r209	3	2022.74	2	1899.20	2	1717.61	2	1563.05	2	1464.66
r210	3	2068.03	2	1913.37	2	1751.23	2	1643.21	2	1520.64
r211	2	1700.50	2	1528.21	2	1384.41	2	1263.33	2	1174.41
Average R2	2.56	1887.72	2.22	1756.50	2.00	1613.97	2.00	1477.30	2.00	1401.46
rc101	14	2740.47	13	2539.38	12	2447.56	12	2296.11	12	2146.39
rc102	12	2451.94	11	2260.62	10	2182.40	10	2007.66	10	1913.18
rc103	10	2203.95	10	2033.79	9	1937.45	9	1828.02	9	1634.42
rc104	9	2077.92	9	1883.60	9	1726.19	9	1546.82	9	1392.22
rc105	13	2674.71	12	2457.12	11	2307.24	11	2128.10	11	2022.89
rc106	11	2287.75	11	2102.13	10	1975.86	10	1798.41	9	1719.77
rc107	10	2197.19	10	2052.82	9	2016.83	9	1738.73	9	1622.31
rc108	10	1977.91	9	1936.41	9	1744.84	9	1583.53	9	1392.21
Average RC1	11.13	2326.48	10.63	2158.23	9.88	2042.30	9.88	1865.92	9.75	1730.42
rc201	4	2939.77	3	2521.89	3	2481.97	3	2371.89	3	2308.24
rc202	3	2459.17	3	2326.46	3	2134.20	3	2024.73	2	1845.68
rc203	3	2103.95	3	1996.00	2	1787.14	2	1729.67	2	1703.96
rc204	2	1861.20	2	1739.90	2	1611.92	2	1486.23	2	1370.88
rc205	3	2586.72	3	2457.34	3	2351.71	3	2285.23	3	2199.26
rc206	3	2253.63	3	2109.32	3	2017.69	2	1777.97	2	1636.20
rc207	3	2119.33	3	2032.39	3	1884.03	2	1708.96	2	1532.61
rc208	3	1884.15	2	1773.21	2	1560.43	2	1443.52	2	1354.49
Average RC2	3.00	2275.99	2.75	2119.56	2.63	1978.64	2.38	1853.53	2.25	1743.92
CNV	397		376		362		354		349	
CNV		239126		225470		213057		200542		189135

Acknowledgments

The author would like to thank Kai-Fabian Henning for his help in coding an earlier version of the algorithm, and Hermann-Josef Kruse, Andreas Stenger and the anonymous referees for their constructive comments on the paper.

Appendix A. Effect of preprocessing rules on VRPTWDST instances

In this section, we investigate the effect of applying the preprocessing rules presented in Section 4.1. To this end, we determine for each problem instance in the base set the percentage of arcs that can be removed due to these rules. As comparison value, the percentage of arcs that can be removed from the underlying Solomon VRPTW instance is also provided. Table 7 presents the results in percent averaged over the instance groups.

The results show that the percentage of infeasible arcs is strongly dependent on the geographical distribution and scheduling horizon of the underlying Solomon group. The existence of driver-specific times only shows a small impact on the percentage: for higher familiarity factors, the percentage of infeasible arcs decreases slightly. This is because some arcs become feasible due to reduced travel and service times. No significant difference between the *Cluster* and *Random* instances can be observed.

Appendix B. Detailed results of TS-DST on VRPTWDST instances

We present detailed results of TS-DST on the base VRPTWDST benchmark set described in Section 5.2.1. TS-DST is run 10 times for each problem instance and we report the number of vehicles (m) and the value of the secondary objective function (traveled distance TD, and working duration WD) of the best solution found. Furthermore, averages over the problem classes C, R and RC and the CNV and the cumulative secondary objective function values (CTD, CWD) are given.

For the secondary objective of minimizing the traveled distance, the results can be found in Table 8 for the instance sets generated with the *Random* mechanism and in Table 9 for the sets generated with the *Cluster* mechanism. Here, we also report CPU times in seconds as the tests were conducted on our desktop PC. For the secondary objective of minimizing working duration, the results are given in Tables 10 and 11. No run-times are reported as the tests were conducted on the computing cluster.

References

Baldacci, R., Battarra, M., & Vigo, D. (2008). Routing a heterogeneous fleet of vehicles. In E. Wasil, S. Raghavan, & B. L. Golden (Eds.), *The vehicle routing problem: Latest advances and new challenges*. In *Operations research/Computer science interfaces: vol. 43* (pp. 3–27). Springer.

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

Bent, R., & Van Hentenryck, P. (2004). A two-stage hybrid local search for the vehicle routing problem with time windows. *Transportation Science, 38*(4), 515–530.

Bettinelli, A., Ceselli, A., & Righini, G. (2011). A branch-and-cut-and-price algorithm for the multi-depot heterogeneous vehicle routing problem with time windows. *Transportation Research Part C: Emerging Technologies, 19*(5), 723–740.

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

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

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

Bräysy, O., Porkka, P. P., Dullaert, W., Repoussis, P. P., & Tarantilis, C. D. (2009). A well-scalable metaheuristic for the fleet size and mix vehicle routing problem with time windows. *Expert Systems with Applications, 36*(4), 8460–8475.

Campbell, A. M., & Thomas, B. W. (2009). Runtime reduction techniques for the probabilistic traveling salesman problem with deadlines. *Computers & Operations Research, 36*(4), 1231–1248.

Cordeau, J.-F., Desaulniers, G., Desrosiers, J., Solomon, M., & Soumis, F. (2002). VRP with time windows. In P. Toth, & D. Vigo (Eds.), *The vehicle routing problem*. In *Monographs on discrete mathematics and applications* (pp. 157–193). SIAM.

Cordeau, J.-F., Gendreau, M., & Laporte, G. (1997). A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks, 30*(2), 105–119.

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

Dell'Amico, M., Monaci, M., Pagani, C., & Vigo, D. (2007). Heuristic approaches for the fleet size and mix vehicle routing problem with time windows. *Transportation Science, 41*(4), 516–526.

Desaulniers, G., Røpke, S., & Madsen, O. (2014). The vehicle routing problem with time windows. In P. Toth, & D. Vigo (Eds.), *Vehicle routing: Problems, methods, and applications, MOS-SIAM monographs on discrete mathematics and applications: vol. 18, Chapter 5* (2nd) (pp. 119–159). Philadelphia, PA, USA: SIAM.

Dondo, R., & Cerdá, J. (2006). An MILP framework for dynamic vehicle routing problems with time windows. *Latin American Applied Research, 36*, 255–261.

Dondo, R., & Cerdá, J. (2007). A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. *European Journal of Operational Research, 176*(3), 1478–1507.

Dongarra, J. J. (2014). Performance of various computers using standard linear equations software. Technical report cs-89-85. *Electrical Engineering and Computer Science Department, University of Tennessee*.

Ferland, J. A., & Michelon, P. (1988). The vehicle scheduling problem with multiple vehicle types. *The Journal of the Operational Research Society, 39*(6), 577–583.

Gendreau, M., Hertz, A., & Laporte, G. (1994). A tabu search heuristic for the vehicle routing problem. *Management Science, 40*(10), 1276–1290.

Gendreau, M., & Potvin, J.-Y. (2010). Tabu search. In M. Gendreau, & J.-Y. Potvin (Eds.), *Handbook of metaheuristics*. In *International series in operations research & management science: vol. 146* (pp. 41–59). Springer.

Gendreau, M., Potvin, J.-Y., Bräysy, O., Hasle, G., & Løkketangen, A. (2008). Metaheuristics for the vehicle routing problem and its extensions: A categorized bibliography. In B. Golden, S. Raghavan, & E. Wasil (Eds.), *The vehicle routing problem: Latest advances and new challenges. Operations research/Computer science interfaces series: vol. 43* (pp. 143–169). Springer.

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

Gillett, B. E., & Miller, L. R. (1974). A heuristic algorithm for the vehicle-dispatch problem. *Operations Research, 22*, 340–349.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research, 13*(5), 533–549.

Goel, A., & Gruhn, V. (2008). A general vehicle routing problem. *European Journal of Operational Research, 191*(3), 650–660.

Groër, C., Golden, B., & Wasil, E. (2009). The consistent vehicle routing problem. *Manufacturing & Service Operations Management, 11*(4), 630–643.

Hashimoto, H., & Yagiura, M. (2008). A path relinking approach with an adaptive mechanism to control parameters for the vehicle routing problem with time windows. *Evolutionary Computation in Combinatorial Optimization*, 254–265.

Hashimoto, H., Yagiura, M., & Ibaraki, T. (2008). An iterated local search algorithm for the time-dependent vehicle routing problem with time windows. *Discrete Optimization, 5*(2), 434–456.

Haughton, M. A. (2008). The efficacy of exclusive territory assignments to delivery vehicle drivers. *European Journal of Operational Research, 184*(1), 24–38.

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

Irnich, S., Schneider, M., & Vigo, D. (2014). Four variants of the vehicle routing problem. In P. Toth, & D. Vigo (Eds.), *Vehicle routing: Problems, methods, and applications. MOS-SIAM series on optimization: vol. 18, Chapter 9* (2nd ed.) (pp. 241–271). Philadelphia, PA, USA: SIAM.

Kindervater, G., & Savelsbergh, M. (1997). Vehicle routing: Handling edge exchanges. In E. Aarts, & J. Lenstra (Eds.), *Local search in combinatorial optimization* (pp. 337–360). John Wiley & Sons.

Kovacs, A. A., Golden, B. L., Hartl, R. F., & Parragh, S. N. (2014a). The generalized consistent vehicle routing problem. *Transportation Science*. doi:10.1287/trsc.2014.0529.

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

Le Bouthillier, A. Le, Crainic, T., & Kropf, P. (2005). A guided cooperative search for the vehicle routing problem with time windows. *IEEE Intelligent Systems, 20*(4), 36–42.

Lim, A., & Zhang, X. (2007). A two-stage heuristic with ejection pools and generalized ejection chains for the vehicle routing problem with time windows. *INFORMS Journal on Computing, 19*(3), 443–457.

Liu, F.-H., & Shen, S.-Y. (1999). The fleet size and mix vehicle routing problem with time windows. *Journal of the Operational Research Society, 50*(7), 721–732.

Nagata, Y., & Bräysy, O. (2009). A powerful route minimization heuristic for the vehicle routing problem with time windows. *Operations Research Letters, 37*(5), 333–338.

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

Pisinger, D., & Røpke, S. (2007). A general heuristic for vehicle routing problems. *Computers & Operations Research, 34*(8), 2403–2435.

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

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

Repoussis, P. P., & Tarantilis, C. D. (2010). Solving the fleet size and mix vehicle routing problem with time windows via adaptive memory programming. *Transportation Research Part C: Emerging Technologies, 18*(5), 695–712.

Repoussis, P. P., Tarantilis, C. D., & Ioannou, G. (2009). Arc-guided evolutionary algorithm for the vehicle routing problem with time windows. *IEEE Transactions on Evolutionary Computation, 13*(3), 624–647.

Røpke, S., & Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science, 40*(4), 455–472.

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

Schneider, M., Doppstadt, C., Sand, B., Stenger, A., & Schwind, M. (2010a). A vehicle routing problem with time windows and driver familiarity. In *Seventh triennial symposium on transportation analysis—Extended abstracts* (pp. 677–680). Tromsø, Norway.

Schneider, M., Doppstadt, C., Stenger, A., & Schwind, M. (2010b). Ant colony optimization for a stochastic vehicle routing problem with driver learning. In *IEEE congress on evolutionary computation* (pp. 1–8). Barcelona, Spain.

Schneider, M., Sand, B., & Stenger, A. (2013). A note on the time travel approach for handling time windows in vehicle routing problems. *Computers & Operations Research, 40*(10), 2564–2568.

Schneider, M., Stenger, A., Schwahn, F., & Vigo, D. (2014). Territory-based vehicle routing in the presence of time window constraints. *Transportation Science*. doi:10.1287/trsc.2014.0539.

Smilowitz, K., Nowak, M., & Jiang, T. (2012). Workforce management in periodic delivery operations. *Transportation Science, 47*(2), 214–230.

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

Soriano, P., & Gendreau, M. (1996). Diversification strategies in tabu search algorithms for the maximum clique problem. *Annals of Operations Research, 63*(2), 189–207.

Sungur, I., Ren, Y., Ordóñez, F., Dessouky, M., & Zhong, H. (2010). A model and algorithm for the courier delivery problem with uncertainty. *Transportation Science, 44*(2), 193–205.

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

Tan, K., Lee, L., Zhu, Q., & Ou, K. (2001). Heuristic methods for vehicle routing problem with time windows. *Artificial Intelligence in Engineering, 15*(3), 281–295.

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

Toth, P., & Vigo, D. (Eds.). (2014). *Vehicle routing: Problems, methods, and applications (2nd ed., vol. 18)*. MOS-SIAM series on optimization. Philadelphia, PA, USA: SIAM.

Vidal, T., Crainic, T., Gendreau, M., & Prins, C. (2015). Time-window relaxations in vehicle routing heuristics. *Journal of Heuristics, 21*, 329–358.

Vidal, T., Crainic, T. G., Gendreau, M., & Prins, C. (2013). A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & Operations Research, 40*(1), 475–489.

Wasner, M., & Zäpfel, G. (2004). An integrated multi-depot hub-location vehicle routing model for network planning of parcel service. *International Journal of Production Economics, 90*(3), 403–419.

Zhong, H., Hall, R. W., & Dessouky, M. (2007). Territory planning and vehicle dispatching with driver learning. *Transportation Science, 41*(1), 74–89.