A Double-Horizon Approach to a Purely Dynamic and Stochastic Vehicle Routing Problem with Delivery Deadlines and Shift Flexibility*

Nikolaus Frohner · Günther R. Raidl

Abstract We are facing a purely dynamic and stochastic vehicle routing problem with delivery deadlines motivated by a real-world application where orders arrive at an online store dynamically over a day to be delivered within short time. Pure dynamism is given since we do not know any orders in advance, whereas the stochastic aspect comes into play by having estimates for the hourly numbers of orders. The goal is to satisfy the daily demand by constructing closed routes from a single depot to the customers given a set of drivers with a predefined shift plan and the hourly demand estimates as input while first minimizing due time violations and then labor and travel costs. Labor costs are subject to optimization since the end times of shifts have a certain amount of flexibility and a decision has to made whether to send home a driver earlier than planned or to extend the shift.

In this work, we present a novel double-horizon approach based on the shifts and the hourly demand estimation. Within the shorter horizon we optimize the routes for the orders currently available whereas within the longer horizon we extrapolate until the end of the day to determine target shift end times for the drivers. Furthermore, we devise a route departure time strategy that balances between route quality and risking due time violations. The routing is performed by a classical adaptive large neighborhood search. We consider artifical instances and compare the results for the online problem with those for the offline scenario where all orders are known from the beginning. We observe superior performance of our approach as compared to fixed route departure time and driver send home strategies.

Keywords Dynamic and stochastic vehicle routing problem · double-horizon approach · adaptive large neighborhood search

* This work is supported by the Vienna Graduate School on Computational Optimization, funded by the Austrian Science Fund under grant W1260.

N. Frohner and G. R. Raidl
Institute of Logic and Computation, TU Wien, Vienna, Austria
{nfrohner|raidl}@ac.tuwien.ac.at

1 Introduction

Motivated by a real-world application where customers place orders at an online store to be delivered within a few hours, we introduce a specific vehicle routing problem (VRP) variant called *Purely Dynamic and Stochastic Vehicle Routing Problem with Delivery Deadlines and Shift Flexibility.* Orders arrive dynamically over the day, and each order is due only a couple of hours after arrival, where the specific due times vary and depend on the orders' types. These orders are picked at a single depot and are subsequently available for delivery to the customers by drivers with predefined shifts.

The goal is to assign the orders to the drivers and perform the routing in a way to avoid or minimize due time violations. Drivers perform multiple routes over the day and for each route a decision has to be made when to start it. This is crucial since after the departure of a driver, the corresponding route cannot be changed anymore. As secondary objective, the labor costs, which are determined by the actual shift end times, and the travel times, determined by the performed delivery routes, are to be minimized. The shift end times are subject to some flexibility and may be ended earlier or extended to account for the uncertainty of the actual load.

In particular, we need to account for the strong dynamism of the problem by making use of the stochastic information known in advance. As such, an estimation of the demand for each hour over the day is available upfront. To link this information to the shifts, the time-dependent average number of orders drivers can handle per hour—the driver performance—needs to be estimated. In this work, we combine well-known adaptive large neighborhood search for vehicle routing [10, 13] with a double horizon approach [8] to handle dynamism and stochasticity. In the short horizon planning, we present a driver performance dependent route departure time strategy—more efficient routes are started earlier than inefficient routes, where improvement is still expected. To avoid sending drivers home too early, we look ahead until the end of the day—the large horizon—by solving a simplified assignment problem on the expected orders without concrete routing to predict target shift end times for the drivers.

In Section 2 we discuss related work. The formalization of the different problem variants (offline, point in time, online), the solution representation, and the objective function is presented in Section 3. Short horizon routes construction is done by adaptive large neighborhood search using classical insertion and regret heuristics and a diverse set of destroy operators as briefly discussed in Section 4. We present the details of our driver performance estimation in Section 5 which is crucial for our departure time strategy (Sect. 6) and our double horizon approach (Sect. 7). The latter is also used to enable informed shift ending strategies as described in Section 8. In the computational study (Sect. 9), we compare the double-horizon approach with fixed, less sophisticated strategies, on artificial instances with different load patterns (business day vs. weekend) and shift plans (generous vs. tight vs. shortage). We observe strong advantages of the former. We conclude in Section 10.

2 Related Work

For general overviews on methods to solve dynamic and stochastic VRPs, see the surveys by Ritzinger and Puchinger [12], Pillac et al. [9], and Psarafitis et al. [11]. Many existing approaches apply periodic or continuous re-optimization of the problem for the current time and essentially ignore information on expected orders. In our context such an approach would not work well as guaranteed delivery times are rather short and started routes cannot be adapted anymore (with respect to the orders they fulfill). Thus, adequately exploiting the estimations of expected orders is of crucial importance.

To handle these uncertainties existing approaches typically fall into one of two categories: those based on *sampling* and those based on *stochastic modeling* [9]. As their name suggests, sampling strategies incorporate stochastic knowledge by generating scenarios based on realizations drawn from suitable random variable distributions. Each scenario is optimized by solving the implied static and deterministic (i.e., offline) problem variant. Then a consensus solution is typically derived from all scenario solutions, which is actually applied in the next time step, until a re-optimization takes place. The advantage of sampling is its relative simplicity and flexibility on distributional assumptions, while its drawback is the massive generation and required solving of scenarios to accurately reflect reality. On the other hand, approaches based on stochastic modeling integrate stochastic knowledge analytically. They try to formally capture the stochastic nature of the problem and are usually highly technical in their formulations and require to efficiently compute possibly complex expected values. Typically, only strong abstractions from the real world allow for stochastic modeling. Applied methods to solve such stochastic models include Markov models and stochastic dynamic programming. In the case of our problem, precise and flexible enough analytical models unfortunately appear to be out of reach.

In the following we review the most relevant existing works we have found in conjunction with our specific VRP.

Bent et al. [3] were one of the first describing an event based model to solve a dynamic VRP. In their *multi plan approach* (MPA) a set of possible routing plans is maintained at any time and updated at certain events. There is one distinguished “best” plan which is determined by an appropriate selection function. The events are new customer requests, vehicle departures according to a current distinguished plan, the availability of newly generated plans, and the timeout of plans. The authors further extend the MPA by sampling to a *multiple scenario approach* (MSA) in order to obtain more robust solutions concerning the stochastic aspects. A number of scenarios is created by adding randomly sampled artificial orders, these scenarios are solved, and then a consensus solution is derived for the original online problem at a certain time. A tabu search is used for actually solving the occurring subproblems. We essentially also follow the fundamental concept of the event based model of the MPA, although with just one current solution.

Hvattum et al. [7] propose another sampling scenario-based approach in conjunction with a rather simple hedging heuristic.

Gendreau et al. [6] describe a tabu search with adaptive memory for a dynamic vehicle routing problem. Essentially, an MPA-like event model is used in conjunction with tabu search and the problem is re-solved whenever new information is available. Stochastic aspects are not considered here, but a focus lies on an effective parallelization.

Ropke and Pisinger [13] and Pisinger and Ropke [10] proposed *Adaptive Large Neighborhood Search* (ALNS) for more general vehicle routing problems, which is nowadays widely used as framework for a large variety of optimization problems. ALNS is appreciated for its practical efficiency as well as robustness on many occasions. The main idea of ALNS is to repetitively destroy a current candidate solution partially and repair it in a sensible way. Both are done by using sets of different basic operators, which are typically randomized. Improved solutions are always accepted as new current ones, while worse solutions are only accepted according to a Metropolis criterion. The application probability of the individual destroy and repair operators are adapted over the iterations based on their successes in previous iterations. ALNS is today among the most often applied metaheuristics for VRPs in general, and we find it also most useful as core optimization technique for solving our routing problem, see Section 4.

Azi et al. [1] consider a VRP with a particular focus on multiple routes per vehicle, as we also have to do. A major difference to our problem is that here the focus is on deciding upon the acceptance of requests. The solution approach is an ALNS that is in several aspects similar to those from Ropke and Pisinger. Azi et al. [1] extend this work towards the dynamic problem variant. Stochastic sampling is applied to account for unknown expected orders.

Schilde et al. [14] describe a variable neighborhood search metaheuristic for a dynamic dial-a-ride problem. The authors also apply sampling for dealing with the stochastic aspects. In their variable neighborhood search the shaking moves bear some similarities with the destroy and repair operations of ALNS.

Mitrović-Minić et al. [8] describe a double horizon approach for solving a dynamic pickup and delivery problem. A large horizon is considered for maintaining routes in a state to be able to easily respond to future dynamically appearing requests, while a short horizon is considered for the actual goal to minimize the route lengths based on the so far known requests. While the considered problem is quite different to ours, we adopt the basic idea of considering two planning horizons in our *double horizon approach* in Section 7.

3 Problem Formalization

We distinguish between three problem variants: the offline problem with full knowledge of the day in advance (OFF), the dynamic problem at a specific time $\tilde{t}$ (DYN-$\tilde{t}$), and the full dynamic problem for a whole day (DYN-DAY).

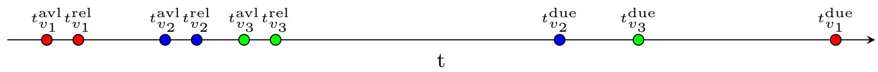

Fig. 1 Visualization of order-related times of an example route $r = \{0, v_2, v_3, v_1, 0\}$. $t^{\text{avl}}_v \leq t^{\text{rel}}_v \leq t^{\text{due}}_v$ holds for all orders: first it is placed by the customer (t^{avl}_v), then it is picked from the warehouse (t^{rel}_v) and ready for delivery by a driver, and then due (t^{due}_v). Note that orders that are placed later may be due earlier. The earliest route departure time is bound from below by the latest release time of the corresponding orders. For this particular example $\tau_r \geq t^{\text{rel}}_{v_3}$ must hold.

3.1 Full-Knowledge Offline Problem (OFF)

Here all orders of the day are known in advance together with their release times, i.e., the times the orders have been picked in the warehouse and are ready for delivery by the drivers. Although this problem variant is not what we are confronted with in reality, it is nevertheless interesting as its (optimal) solution provides a baseline of what might ideally be achieved in the online problem. We denote the set of all orders by V, with $n = |V|$, and the corresponding release times by t^{rel}_v, $\forall v \in V$. Moreover, we are given due times t^{due}_v, $\forall v \in V$, which are related to a promised maximum delivery duration starting from the time t^{avl}_v the order v was placed by the customer. Fig. 1 visualizes the order-related times of an example route consisting of three orders.

Furthermore, for all relevant vehicles $u \in U$, with $m = |U|$, planned shift time intervals $[q^{\text{start}}_u, q^{\text{end}}_u]$ and earliest shift ends $q^0_u \in [q^{\text{start}}_u, q^{\text{end}}_u]$ are provided. Lastly, expected travel times $\delta(v, v')$ from location v to location v', where $v, v' \in V \cup \{0\}$ with 0 representing the warehouse, are given. These travel times include average stop times at the customers, average times for loading a vehicle at the warehouse, and postprocessing times when returning to the warehouse. We further assume that the triangle inequality holds w.r.t. the travel times and that they are constant throughout the day.

Solution Representation. We have to plan the drivers' routes, the route departure times, and the drivers' flexible shift end times. Hence, a candidate solution is a tuple $\langle R, \tau, q \rangle$ where

- $R = (R_u)_{u \in U}$ denotes the *ordered sequence of routes* $R_u = \{r_{u,1}, \ldots, r_{u,\ell_u}\}$ to be performed by each vehicle $u \in U$, and each *route* $r \in R_u$ is an ordered sequence $r = \{v^r_0 = 0, v^r_1, \ldots, v^r_{l_r}, v^r_{l_r+1} = 0\}$ with $v^r_i \in V$, $i = 1, \ldots, l_r$, being the i-th order to be delivered and 0 representing the warehouse at which each tour starts and ends,
- $\tau = (\tau_r)_{r \in R_u, u \in U}$ are the (planned) *departure times* of the routes, and
- $q = (q_u)_{u \in U}$ are the *shift end times* of the vehicles.

The time at which the i-th order v^r_i of route r, $i = 1, \ldots, l_r$, is delivered is

$$a(r, i) = \tau_r + \sum_{j=0}^{i-1} \delta(v^r_j, v^r_{j+1}). \tag{1}$$

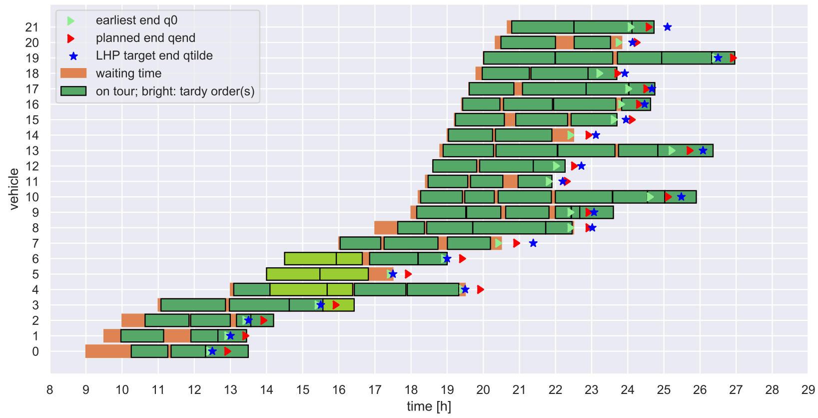

Fig. 2 Visualization of a solution for an artificial instance with 22 vehicles. The x-axis denotes the time and the discrete y-axis the drivers' shifts. The whole bar indicates the actual shift duration. The green triangle indicates the earliest shift end time for each driver, where excess labor time contributes to our considered costs. The red triangle depicts the planned shift ending, after which no route can be started, but the last route may end arbitrarily late. The distinct green bars stand for routes and contribute to the travel time part of our objective function. If the latter is shaded light green, the route contains at least one tardy order, which can be observed around hour 15. The remaining orange of the shift bars denote waiting time of the driver at the depot. The blue stars denote the target end shift times as determined by the large horizon planning in the beginning of the day.

The total duration of a route $r \in R_u$ of a vehicle $u \in U$ is

$$d(r) = \sum_{i=0}^{l_r} \delta(v_i^r, v_{i+1}^r), \tag{2}$$

and the route therefore is supposed to end at time $\tau_r + d(r)$.

Let $\tau^{\min}(r) = \max_{i=1,\ldots,l_r} t^{\text{rel}}_{v_i^r}$ be the earliest feasible starting time of a route r, which corresponds to the maximum release time of the orders served in the route. Furthermore, let $\tau^{\max}(r)$ be the latest starting time without violating any due time, i.e.,

$$\tau^{\max}(r) = \min_{i=1,\ldots,l_r} \left(t^{\text{due}}_{v_i^r} - \sum_{j=0}^{i-1} \delta(v_j^r, v_{j+1}^r) \right). \tag{3}$$

Feasibility. A solution is feasible when

- each order $v \in V$ appears exactly once in all the routes in $\bigcup_{u \in U} R_u$,
- each route $r \in R_u$, $u \in U$, is started in the planned shift time of the assigned vehicle, i.e., $\tau_r \in [q_u^{\text{start}}, q_u^{\text{end}}]$,
- and not started before all corresponding orders are released, i.e., $\tau_r \geq \tau^{\min}(r)$,
- the routes in each R_u, $u \in U$ start at increasing times and do not overlap, i.e., $\tau_{r_{u,i}} + d(r_{u,i}) \leq \tau_{r_{u,i+1}}$, $i = 1, \ldots, |R_u| - 1$,

– and the actual shift end time is not smaller than the finishing time of the last route (if there is one) and the minimum shift time, i.e., $q_u \geq \max(q_u^0, \sup_{r\in R_u}(\tau_r + d(r))),\ u \in U$.

Objective. The primary goal is to avoid tardiness or distribute it evenly among the customers. The secondary goal is reduce labor and travel costs. This leads to the following objective function to be minimized

$$f(\langle R,\tau,q\rangle) = \mathrm{L}\left(\sum_{r\in R_u, u\in U}\sum_{i=1}^{l_r}\max(0, a(r,i) - t^{\mathrm{due}}_{v_i^r})^2,\ \gamma\cdot\sum_{u\in U}(q_u - q_u^0) + \sum_{r\in R_u, u\in U} d(r)\right). \tag{4}$$

L denotes the lexicographic combination of two terms, which are a quadratic penalty for the tardiness of deliveries and a linear combination of the sum of labor and travel costs. More precisely, the latter is calculated as the sum of the actual shift durations above the minimum shift times q_u^0 weighted by a factor γ and the sum of travel times.

In a real-world comparison of results, it is also worthwhile to view it as a multi-objective optimization problem. A small increase in tardiness may be acceptable, if it comes with a substantial reduction of costs.

3.2 Dynamic Problem at a Specific Time $\tilde{t}$ (DYN-$\tilde{t}$)

This problem variant is actually the one that needs to be iteratively solved during the whole day, for increasing current time $\tilde{t}$. It extends OFF by having as additional input the current time $\tilde{t}$ and the expected number of orders $\hat{\omega}(t)$ that become available in the time intervals $[t, t+1\mathrm{h})$ for all relevant business hours. Moreover we assume to have knowledge about the distribution of order types w.r.t. the promised delivery durations. The set of all orders V is reduced to those which are already available at time $\tilde{t}$ and whose delivery has not yet started. The set of vehicles U is reduced to those whose shift has not been finished, and shift start times are updated to expected return times of vehicles that are currently on a tour. The route construction must now additionally consider these unknown future orders in an appropriate way. The ultimate goal is to lead to an optimal solution w.r.t. the full dynamic problem below.

3.3 Full Dynamic Problem (DYN-DAY)

This is the actual problem to be solved from the point-of-view of the whole day. Time is considered to continuously increase over the whole relevant time horizon, expected numbers of future orders are known as above, but each concrete order becomes available only at the availability time $t_v^{\mathrm{avl}},\ \forall v \in V$. The decision on each route $r \in R$ must be fixed with only the knowledge available up to the routes respective departure time τ_r. An example solution of a DYN-DAY instance with 22 vehicles is depicted in Fig. 2 as a bar chart displaying the waiting times (orange), routes (green), and routes with tardy orders (light green) of the drivers. Stars show the target shift end times derived from our initial large horizon planning (Sect. 7).

More specifically, we solve the successive DYN-$\tilde{t}$ instances every time an order is released:

$$\tilde{t} \in \left\{t \mid \exists v \in V : t = t_v^{\mathrm{rel}}\right\} \tag{5}$$

Having obtained a solution for a time $\tilde{t}$, we extract any routes that start before the next value for $\tilde{t}$ in the above sequence, adopt these routes for the final solution of DYN-DAY, and remove all the orders served in these routes from any further consideration.

4 Routes Construction and Optimization

To be suitable for a real-time application, an important property that an optimization method must exhibit is a good *anytime behavior*: a somehow reasonable heuristic solution must be found very soon (within seconds), and over time the solution should continuously be improved up to (or close to) optimality. In other words, the optimization can be interrupted almost at any time and a reasonable solution with respect to the invested time is available. We achieve this by using a carefully designed *Adaptive Large Neighborhood Search* (ALNS) [9,13].

ALNS heuristics. As construction heuristics to insert orders into either an empty solution or to repair a partial solution in the ALNS, we use the well-known insertion and regret-k heuristics as described in [10]. We distinguish between the *zero-tardiness* and *tardiness* regimes. In a two-stage approach, we first seek to insert an order without introducing additional tardiness, which can be checked in constant time with caching of suitable slack values for existing orders and routes. If this is not possible, we search for an order position with the smallest sum of squares increase of tardiness, which is computationally more demanding by a factor of $O(n)$.

Our destruction heuristics are mostly adopted from Pisinger and Ropke [10], Ropke and Pisinger [13], Shaw [15], and Azi et al. [1] and suitably adapted to our problem. There are two kinds of destruction heuristics, those that remove a certain number of orders from routes and those that remove a certain number of whole routes. More specifically, we use *random order*, *random route*, *related order*, *related route*, *worst order*, and *worst and related order* removal.

Shift End Times. The actual shift end times q_u for the vehicles are set to $\max(q_u^0, \sup_{r \in R_u}(\tau_r + d(r)))$, i.e., for each vehicle u to the end of the last route or the earliest possible shift end time, whichever comes later. In Section 7, we introduce the large horizon planning, where we estimate desired shift ends for the vehicles in advance so that we can satisfy the expected workload. Since in the objective function we penalize labor time after the earliest possible shift end times q^0, we grant vehicles that are below their desired shift end time $\tilde{q}_u > q_u^0$ a labor time bonus that equalizes the incurred labor time costs up until $\tilde{q}_u$—otherwise, the insertion heuristic would avoid assigning orders to vehicles after their earliest possible shift end q_u^0, in case there is no tardiness yet and other vehicles not close to their shift end are available. The labor time bonus is implemented by using the augmented objective function

$$\tilde{f}(\langle R, \tau, q \rangle)) = f(\langle R, \tau, q \rangle) - \gamma \cdot \sum_{u \in U} \min(q_u - q_u^0, \tilde{q}_u - q_u^0). \tag{6}$$

During the optimization, the route departure time is always set to the earliest possible time. Afterwards, we are free to postpone the routes up to the latest time within the departure time slacks of the routes so that the objective value is neither increased by tardiness nor by labor costs.

5 Driver Performance Estimation

For both an informed route departure time strategy and our large horizon approach, we need to estimate the driver performance of a given hour. It is the average time needed to serve an order. It is strongly related to the expected duration of all routes involved to serve the customers at the considered time interval divided by the number of customers. We introduce this as a function $\phi\colon \mathbb{R} \to \mathbb{R}$, depending on the load λ. We define the load λ to be the expected number of orders due in a given hour.

A classical result by Beardwood et al. [2] shows that the expected length of an optimal traveling salesperson tour with n randomly sampled cities given some geometry with area $\mathcal{A}$ grows with $k\sqrt{\mathcal{A}n}$. k is an empirical constant depending on the spatial distribution and the metric. This result is extended to capacitated vehicle routing problems by Daganzo [4] and refined by Figliozzi [5], from which we adapt the following model to explain $\phi(\lambda)$

$$\phi(\lambda; k_m, k_l) \approx k_m + \frac{k_l}{\sqrt{\lambda + 1}}. \tag{7}$$

k_m corresponds to constant costs occurring for each customer like the stop time at the customer. k_l relates to the empirical k from [2] and accounts together with $(\lambda + 1)^{-1/2}$ for the expected travel time to a customer. We shift the load by one to avoid divergence at zero load. As we can see, it is a function that decreases with the square root of the load. As a more flexible model, we further suggest the following inverse power law

$$\phi(\lambda; k_m, k_l, \alpha) \approx k_m + \frac{k_l}{(\lambda + 1)^\alpha}. \tag{8}$$

To check the validity of these models in our setting and tune the parameters, we create ten artificial instances each for loads starting from 0.5 up to 20 in steps

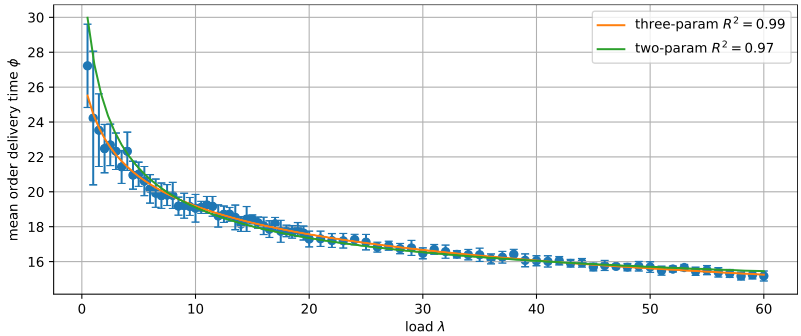

Fig. 3 Mean order delivery times ϕ in minutes with standard errors over ten instances for each load value $\lambda \in \{0.5, 1.0, \ldots, 20, 21, \ldots, 60\}$ with fitted curves for the two and three-parameter models. The three-parameter model seems to explain the region of little load $\lambda \leq 10$ better than the two-parameter model.

of 0.5 and further in steps of 1 up to 60, i.e., to one order due per minute. The geometry is the unit disk with a central depot, Euclidean metric, and vehicles driving a constant pace of 20 minutes per unit distance. Furthermore, constant stop times at the customers, loading times when leaving the warehouse, and postprocessing times when returning to the warehouse are added. Orders arrive randomly throughout a whole day at a given constant rate λ sampled from a Poisson process following a uniform spatial distribution. Optimization at each DYN-$\tilde{t}$ is done for 60 seconds using ALNS. Sufficient drivers are available so that no tardiness occurs, and the drivers wait to start their routes as long as possible. For each instance, we average over all routes the time needed to serve a customer.

In Fig. 3 we see a scatter plot of the mean order delivery times including standard errors ($N = 10$) over the different loads. Weighted least squares fits of the models are displayed. Both models explain the data starting from load $\lambda \geq 10$ similarly well with weighted R^2 values of 0.97 and 0.99. For low-load regions $\lambda \leq 10$, the model with the inverse power as an arbitrary parameter lies closer to the means.

6 Departure Time Strategies

In the dynamic problem, at every time $\tilde{t}$ we construct and optimize the routes for the drivers. After that, we have to decide when these should be started. A departure time window $[\tau_r^{\text{earliest}}, \tau_r^{\text{latest}}]$ is attributed to each route, within which the departure time τ_r of the route may be set while maintaining a feasible solution and not increasing the objective value. Setting $\tau_r < \tau_r^{\text{earliest}}$ or $\tau_r > \tau_r^{\text{latest}}$ makes the solution either infeasible or increases tardiness, labor, or travel costs. This decision is crucial since routes cannot be adapted anymore after they have been started.

Two naive strategies can immediately be devised by either always starting the route at τ^{earliest} or at τ^{latest}. τ^{latest} seems favorable in most situations since not yet started routes may later still be adapted in order to more efficiently include newly emerged orders as opposed to the earliest strategy where in the extreme case a route is immediately started with just one order. However, experiments have shown that the start-latest strategy is not always the better strategy, since we may run into tardiness at a later time when working at or shortly before critical utilization and letting vehicles wait instead of delivering orders.

A more sophisticated approach takes into account the current performance of a route, measured by its number of minutes per order d_r^{O}, i.e., the route duration divided by the number of orders served. The main idea is: the better the performance of a route, the closer we can set its departure time τ_r towards τ_r^{earliest}, the worse, the closer towards τ_r^{latest}, so that there is a performance-dependent time for improvement by further incoming orders. As we have seen in detail in Section 5, the performance depends on the load by an inverse power law.

We assume a Gaussian distribution of $d_r^{\text{O}} \sim \mathcal{N}(\phi(\lambda), \sigma_\phi^2(\lambda))$ and set the departure time of a route to

$$\tau_r(d_r^{\text{O}}, \lambda) = \tau_r^{\text{earliest}} + (\tau_r^{\text{latest}} - \tau_r^{\text{earliest}}) \cdot \Phi\left(\frac{d_r^{\text{O}} - \phi(\lambda)}{\sigma_\phi(\lambda)}\right), \qquad (9)$$

where Φ is the cumulative normal distribution function. For example, when d_r^{O} corresponds exactly to the expected mean order delivery time $\phi(\lambda)$ in the given load situation, the departure time of r will be set to $(\tau_r^{\text{earliest}} + \tau_r^{\text{latest}})/2$, the middle of the route departure time slack. We estimate $\sigma_\phi(\lambda)$ by calculating sample standard deviations from our experiments described in the previous section.

We will refer to the three different strategies as τ-earliest, τ-latest, and τ-route.

7 Double Horizon Approach

This approach adopts from Mitrović-Minić et al. [8] the idea of considering in the optimization two planning horizons simultaneously, a short horizon and a large horizon. In the *Large Horizon Planning* (LHP), which we always perform as first step, we consider a strongly simplified approximate problem variant of DYN-$\tilde{t}$ where, in addition to all available requests, also all the expected future requests for either the whole day or at least several hours into the future. The primary goal is to make a rough plan on the utilization of the vehicles and recognize times where we might exceed the available capacity or have enough time to finish vehicle shifts earlier. A detailed routing is *not* done in the LHP. The short horizon problem corresponds to our definition of DYN-$\tilde{t}$ so far but utilizes an adapted objective function that includes additional terms defined by the LHP's results in order to meet the long-term goals as closely as possible.

In our case decisions on the labor time to be used beyond the minimum q_u^0 for each vehicle are most critical in the long-term in order to avoid later deliveries becoming tardy due to insufficient driving resources for the given workload.

We therefore define and solve the following LHP subproblem at time $\tilde{t}$ in order to derive *target shift end times* $\tilde{q}_u$ for each vehicle $u \in U$. We consider as V all currently relevant orders of the current DYN-$\tilde{t}$ plus expected orders V^{exp} for the remaining day. These expected orders are artificially created according to the estimated numbers of orders becoming available per hour $\hat{\omega}(t)$, equidistantly spaced over each hour. For each of these orders we further derive a due time randomly based on the distribution of expected order types and their promised maximum delivery times.

Let $z\colon \mathbb{R}^+ \to \mathbb{R}^+$ be a function that estimates the average shift duration needed to serve one order $v \in V \cup V^{\mathrm{exp}}$ within the current hour of $\tilde{t}$ and a few subsequent hours, assuming a reasonable routing and an average number of available orders. The basis for z is the mean order delivery time $\phi(\lambda)$ derived from the routes with latest departure time strategy as presented in Section 5. To account for a slight increase due to waiting times in the depot and an intermediate departure time strategy, we introduce an additional factor $\zeta \gtrapprox 1$. With $\Lambda(t)$ being the load at hour t, we then calculate a weighted average to estimate the average shift duration

$$z(\tilde{t}) = \zeta \cdot \frac{\sum_{t'=\tilde{t}}^{\tilde{t}+\rho} \Lambda(t') \cdot \phi(\Lambda(t'))}{\sum_{t'=\tilde{t}}^{\tilde{t}+\rho} \Lambda(t')}, \tag{10}$$

with ρ corresponding to three hours in our implementation. We make the strongly simplifying assumption that any order v can be independently served by any available vehicle within time $z(\tilde{t})$ from t_v^{rel} onward. Each vehicle's shift is split into successive time slots of duration $z(\tilde{t})$, and in each of these time slots one order can be served. This implies that we do not allow arbitrary start times to serve orders but only times that are multiples of $z(\tilde{t})$ away from a vehicles' shift start time (or $\tilde{t}$). We do not have a strict last slot, i.e., in principle further orders to be served might always be appended to a vehicles shift. An instructive visualization of the LHP's view on an example DYN-$\tilde{t}$ instance is provided in Figure 4.

A solution to our LHP is a complete assignment of all the orders $V \cup V^{\mathrm{exp}}$ to vehicle slots. As actual delivery time of an order we consider the respective time slot's middle point, i.e., the time slot's start time plus $z(\tilde{t})/2$. The objective function corresponds to our main objective function (4), but as we do not consider routing the last travel time term is omitted.

This LHP is heuristically solved by a greedy assignment procedure, in which orders are assigned in increasing due-time always to the earliest feasible time slot of a vehicle that increases the objective the least. In case of ties, a vehicle $u \in U$ whose end of the shift exceeds q_u^0 the least, i.e., where the vehicle's excess labor time is smallest, is chosen. This aspect automatically balances the deviations from the planned shift times among the vehicles if

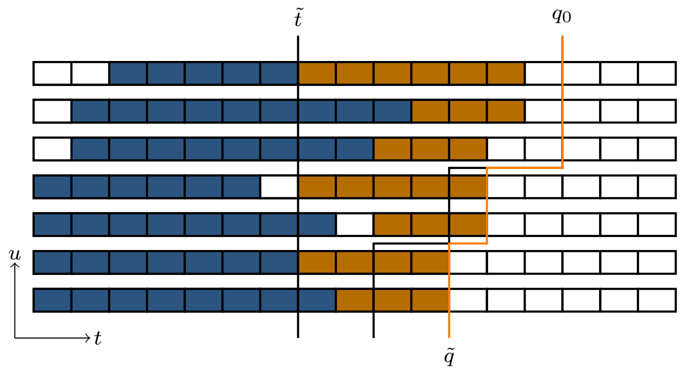

Fig. 4 View on an example DYN-$\tilde{t}$ problem instance as seen by the large horizon planning. The x-axis represents the time, discretized by time slots of the average expected shift time needed to deliver an order $z(\tilde{t})$. The drivers that are still available at or after $\tilde{t}$ are stacked on the y-axis. Blue rectangles indicate orders that have already been delivered or are en route. Brown rectangles represent greedily assigned orders, either real (available at the moment) or expected up until the end of the planning horizon. The maximum of the earliest shift end time q_u^0 and the latest assigned order define for each driver the target shift end time $\tilde{q}_u$. For the last four drivers q_0 is exceeded, since otherwise tardiness would have arisen. Note that unassigned slots may occur if no more orders are ready for delivery at that time.

there are no particular other reasons such as avoiding tardy orders. Further ties are resolved randomly by a random processing order of the vehicles.

The obtained shift end times of this solution, i.e., the end times of the last used time slots of each vehicle, are finally used as target shift end times $\tilde{q}_u$, for all $u \in U$ in the short horizon optimization, i.e., the ALNS from Section 4.

This is achieved by augmenting objective function (4) to

$$\tilde{f}(\langle R, \tau, q \rangle)) = f(\langle R, \tau, q \rangle) + \gamma \cdot \sum_{u \in U} Q_u \tag{11}$$

with

$$Q_u = -\min(q_u - q_u^0, \tilde{q}_u - q_u^0). \tag{12}$$

This non-positive term can be seen as bonus that exactly compensates any arising labor time costs above q_u^0 up to the target time $\tilde{q}_u$ for each vehicle $u \in U$. Thus, the time up to $\tilde{q}_u$ can be used "for free". Note that the factor γ by which the bonus is multiplied is the same as by which the labor time is weighted in (4).

8 Shift Ending Strategies

In the online problem, we also have to decide if a shift should be ended by sending a driver (vehicle) $u \in U$ home, providing this is allowed, i.e., $\tilde{t} \geq q_u^0$, u is in the depot, and no more routes are planned for u, or if the driver has to wait at the depot to possibly receive further orders. Again, two naive strategies are immediately available: The first option is to send a vehicle u home as early as possible, i.e., after its last so far planned route or at q_u^0, whichever comes later. This is also the default of the insertion heuristic. The other extreme is the latest strategy that waits until q_u^{end} in any case, even if the last route ends before q_u^{end}. The earliest strategy seems to be an attractive choice, since we can save labor costs and during peak hours, it is likely that a vehicle has already a next route planned during its current route, therefore it is not sent home prematurely when arriving at the depot, if there is still enough work to do.

A more sophisticated approach makes use of the estimated shift end times $\tilde{q}$ provided by the LHP. The earliest shift end time is then modified to be $\tilde{q}_u - \tilde{d}$, where $\tilde{d}$ is a threshold duration of an efficient route. The rationale is that if a vehicle cannot start a somewhat efficient route that ends before its target shift end $\tilde{q}_u$, it is better to send it home.

We will refer to the three different strategies as q-earliest, q-latest, and q-LHP.

9 Computational Study

We conducted all our experiments on Intel Xeon E5-2640 processors with 2.40 GHz in single-threaded mode and a memory limit of 8 GB. We implemented our approach as a prototype in Python 3.7, being aware that an implementation in a compiled language would be substantially faster and have a smaller memory footprint. We consider six different instance classes, each with 20 instances: Artificial instances[1] on the Euclidean unit disk as described in Section 5 using either a business day (BD) or a weekend (WE) load profile with generous (GE), tight shift planning (TI), and with a shortage (SH) of drivers. The idea is to observe the transition from a more generous shift planning to a tighter one and simulating a driver being absent on short notice where in the latter cases more tardiness is expected to occur. Furthermore, in the generous case, dynamically ending shifts earlier is expected to have more impact where in the tight case shifts are more likely to be extended by starting long routes shortly before the ending.

We aim at comparing the performance of the naive earliest and latest strategies with the more sophisticated LHP and driver performance based route departure strategy on those DYN problem instances. In each case, we apply the ALNS with a limit of 1000 non-improving iterations and additionally a 60 seconds time limit for route optimization at each arriving order. This should be consistent with a real-time setting, where orders may arrive every minute during peak-time or on weekends and routes already including them should occasionally be started within a minute. Without LHP and driver performance estimation, we are restricted to naive earliest and latest strategies regarding the departure time of a route and the early shift termination. LHP extrapolates until the end of the horizon to set desired shift ending times for

[1] https://github.com/nfrohner/pdsvrpddsf

Algorithm 1: Simulated DYN Problem Solver with ALNS and LHP.

Input: Orders V, drivers U, shift starts q^{start}, earliest shift ends q^0, planned shift ends q^{end}, hourly expected number of orders $\hat{\omega}$, travel time matrix δ, mean order delivery time ϕ, efficient route threshold duration $\tilde{d}$.
Output: Solution $\langle R, \tau, q\rangle$ with routes R, route departure times τ, and actual shift end times q for the whole day.

$V^{\text{deliv}} \leftarrow \{\}$;
$U^{\text{home}} \leftarrow \{\}$;
$R' \leftarrow ()_{u\in U}, \tau' \leftarrow ()_{r\in R'}, \tilde{t}' \leftarrow 0, \tilde{q} \leftarrow q^{\text{end}}$;
$\langle R, \tau, q\rangle \leftarrow \langle R', \tau', q^0\rangle$;
foreach $\tilde{t} \in \{t \mid \exists v \in V : t = t_v^{\text{rel}}\} \cup \{\infty\}$ **do**
  foreach $(u, r) \in R' : \tilde{t}' \le \tau_r' < \tilde{t}$ **do**
    $V^{\text{deliv}} \leftarrow V^{\text{deliv}} \cup \{v_1^r, \ldots, v_{l_r}^r\}$;
    $\langle R, \tau, q\rangle \leftarrow \langle R, \tau, q\rangle \oplus (r, \tau_r')$;
    $q_u^{\text{start}} \leftarrow u$'s return time at depot after $\tilde{t}$;
  end
  $U^{\text{home}}, q \leftarrow \text{SENDHOME}(\tilde{t}, \tilde{t}', U^{\text{home}}, U \setminus U^{\text{home}}, q^{\text{start}}, q^0, q^{\text{end}}, \tilde{q}, \tilde{d})$;
  $V^{\text{avl}} \leftarrow \{v \in V \setminus V^{\text{deliv}} : t_v^{\text{avl}} \le \tilde{t}\}$;
  $\tilde{q} \leftarrow \text{LHP}(\tilde{t}, V^{\text{avl}}, U \setminus U^{\text{home}}, \hat{\omega}, \phi, q^{\text{start}}, q^0, q^{\text{end}})$;
  $\langle R', \tau', q\rangle \leftarrow \text{ALNS}(\tilde{t}, V^{\text{avl}}, U \setminus U^{\text{home}}, q^{\text{start}}, q^0, q^{\text{end}}, \tilde{q}, \delta)$;
  $\tau' \leftarrow \text{DEPART}(R', \phi)$;
  $\tilde{t}' \leftarrow \tilde{t}$;
end
return $\langle R, \tau, q\rangle$;

each driver, using an estimation of the average driver performance in the window of the current and the upcoming three hours. The target shift ending times may be before the planned shift ends to send drivers home early or after them so that extending shifts is favored via the augmented objective function.

In Algorithm 1 we list a high-level pseudo-code of the simulated DYN problem solver, combining the previously explained approaches based on the LHP and route performance. The main loop goes over all times $\tilde{t}$ where an order is released, where the first inner loop checks whether routes have been started between the last and the current $\tilde{t}$. If so, they are added to the current solution, the corresponding orders are removed, and the drivers' shift starts are set to their return times at the depot. Afterwards, drivers are sent home, if their target shift end time reduced by the efficient route threshold $\tilde{q} - \tilde{d}$ passed and they have no further routes planned. Then the route construction and optimization begins with the large horizon planning to update the $\tilde{q}$. It further continues with the ALNS—the optimization workhorse—that creates routes for the currently available and not yet delivered orders. Finally, the departure time of the planned routes is set according to the route performance strategy (more efficient routes are planned to start earlier).

In Table 1, we present the main results of our computational study. We compare for the different combinations of our approaches means and standard deviations of the number of tardy orders n^{tardy}, the root mean squared error

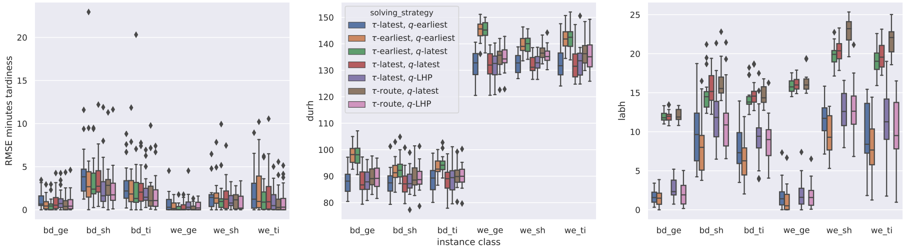

Fig. 5 Comparison of the root mean square error of the tardiness in minutes, the travel time duration, and the excess labor time of different solution strategies (without offline solution) on six different instances classes with 20 instances each. We observe that the more sophisticated strategies based on LHP and the route performance decreases the tardiness at the cost of carefully introducing additional travel time (regarding which τ-latest is best) and labor time (where q-earliest is best).

(RMSE) of the tardiness in minutes, the total travel time in hours, the labor time exceeding q_u^0 in hours, the average route duration $\bar{d}$, and the average route performance (labor time to serve an order without waiting time) in minutes d_r^{O}. The offline (full knowledge of the day) results (OFF) where we applied ALNS to convergence with a limit of 1000 non-improving iterations without additional time limit provide a performance baseline. All the other results are for the DYN-DAY problem variant and we see that the offline baseline is somewhat out of reach, which is not too surprising due to the substantially restricted knowledge that can be exploited in the online variant. Despite having used a lexicographic optimization approach, where distributing tardiness evenly and reducing it was the single most important objective, we analyze the results in the sense of a multi-objective optimization problem. Small amounts of tardiness for a few customers may in practice be acceptable when real costs may be substantially reduced. In Figure 5, we visualize the results by boxplots of the tardiness, travel time, and labor time, for the different solution strategies (excluding the offline problem) on the six different instance classes.

We observe that the τ-latest strategy provides the best route performances and therefore smallest travel costs but sometimes runs into troubles regarding tardiness, where a τ-earliest strategy would have been beneficial. Similarly, the q-latest strategy provides the most shift time resources allowing to reduce tardiness, as opposed to the q-earliest strategy. The goal of τ-route is to balance between the extremes of the τ determination strategies considering the load development of the day. Likewise, the q-LHP strategy should provide additional shift resources regarding the flexibility of shift endings times only when necessary. We observe that the τ-route strategy sacrifices a slight amount of route quality in exchange for substantially less tardiness. Likewise, the LHP carefully provides additional labor time to be used to reduce tardiness. Combining both strategies results in a reasonable trade-off over all the instances classes, where a decision maker may also select a suitable combination of strategies given the load and shift structure of the day.

Table 1 Offline problem performance (OFF) and different solution strategies applied to 20 artificial instances for each configuration using either a business day (BD) or a weekend (WE) load profile with either generous, tight, or shift planning with a driver shortage.

			n^{tardy}		RMSE [min]		dur [h]		lab [h]		$\bar{d}$ [min]		d_r^{O} [min]	
			mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
load	shift	solving strategy												
BD	generous	τ-earliest, q-earliest	4.500	4.536	0.680	0.798	98.134	3.580	1.491	1.069	56.425	4.315	18.386	0.692
		τ-earliest, q-latest	3.850	4.771	0.592	0.795	98.087	3.984	11.940	0.698	55.930	3.707	18.372	0.631
		τ-latest, q-LHP	6.250	6.257	1.107	1.208	88.221	4.447	2.685	1.199	74.935	2.832	16.512	0.426
		τ-latest, q-earliest	7.600	6.065	1.157	1.091	87.716	4.443	1.673	0.889	74.745	2.818	16.420	0.479
		τ-latest, q-latest	5.900	5.937	1.028	1.237	87.980	5.210	11.974	0.722	75.460	2.792	16.463	0.574
		τ-route, q-LHP	4.400	4.604	0.841	1.134	89.369	4.568	2.197	1.560	71.340	2.113	16.728	0.483
		τ-route, q-latest	4.350	4.782	0.757	1.082	89.956	4.241	12.036	0.797	70.060	2.398	16.839	0.379
		OFF	0.550	1.572	0.056	0.174	78.897	4.763	0.377	0.479	64.170	2.039	14.760	0.384
	shortage	τ-earliest, q-earliest	19.950	16.804	3.991	4.966	91.863	4.294	8.334	3.843	63.415	4.985	17.259	0.468
		τ-earliest, q-latest	19.100	13.726	3.381	2.568	92.763	4.305	14.775	2.426	62.930	4.941	17.430	0.533
		τ-latest, q-LHP	14.800	10.928	2.705	2.915	88.099	4.742	12.262	3.854	74.640	3.749	16.544	0.350
		τ-latest, q-earliest	26.200	14.207	4.179	2.747	87.507	5.044	9.946	4.296	74.355	4.007	16.430	0.370
		τ-latest, q-latest	22.737	16.562	3.551	2.820	87.524	5.038	15.406	2.073	72.126	3.019	16.505	0.426
		τ-route, q-LHP	15.450	11.019	2.546	2.525	89.483	4.717	11.454	4.103	71.270	3.763	16.806	0.388
		τ-route, q-latest	18.150	11.864	2.988	2.338	89.992	4.691	16.304	2.145	70.045	3.622	16.903	0.375
		OFF	2.050	5.826	0.255	0.721	81.655	4.358	4.671	3.230	57.595	1.676	15.336	0.400
	tight	τ-earliest, q-earliest	14.158	10.673	2.720	2.928	93.582	3.530	6.319	2.803	62.416	4.261	17.507	0.746
		τ-earliest, q-latest	10.950	8.003	2.847	4.480	94.496	3.263	14.236	1.601	61.630	3.886	17.670	0.725
		τ-latest, q-LHP	14.300	11.855	2.139	1.923	89.265	5.296	9.496	3.458	73.180	2.846	16.667	0.477
		τ-latest, q-earliest	17.150	9.544	2.959	2.341	88.868	5.074	7.920	2.934	72.865	2.211	16.593	0.457
		τ-latest, q-latest	13.900	9.754	2.466	2.179	88.732	4.822	14.810	1.636	70.945	2.330	16.571	0.514
		τ-route, q-LHP	8.950	6.117	1.843	2.098	90.030	4.238	8.883	2.960	70.370	2.580	16.818	0.452
		τ-route, q-latest	10.105	7.880	2.028	2.542	90.265	4.403	14.728	1.231	68.816	1.937	16.870	0.509
		OFF	0.700	2.904	0.115	0.512	82.371	5.380	3.273	2.645	58.400	2.307	15.374	0.420
WE	generous	τ-earliest, q-earliest	2.400	3.267	0.420	0.539	145.071	3.794	1.213	1.660	62.140	3.102	17.008	0.504
		τ-earliest, q-latest	1.750	2.900	0.273	0.435	145.130	3.561	15.889	0.853	61.985	3.286	17.021	0.634
		τ-latest, q-LHP	2.850	2.661	0.701	1.056	132.071	4.851	2.024	1.885	76.835	2.059	15.477	0.343
		τ-latest, q-earliest	2.850	2.978	0.745	1.112	132.159	5.094	1.715	1.753	76.580	2.008	15.486	0.334
		τ-latest, q-latest	1.600	2.137	0.358	0.531	132.161	4.962	16.099	0.859	76.015	1.727	15.488	0.335
		τ-route, q-LHP	2.450	2.837	0.455	0.551	134.841	4.713	1.820	1.657	73.420	2.295	15.800	0.329
		τ-route, q-latest	1.650	2.641	0.271	0.411	134.682	4.460	16.286	1.248	73.945	1.934	15.787	0.435
		OFF	0.000	0.000	0.000	0.000	118.698	5.720	0.197	0.407	67.455	2.253	13.902	0.291
	shortage	τ-earliest, q-earliest	12.950	9.801	1.743	1.817	139.469	3.321	9.636	2.564	67.160	3.392	16.167	0.597
		τ-earliest, q-latest	7.400	5.623	1.475	2.305	139.888	3.335	19.844	1.440	66.135	2.164	16.215	0.571
		τ-latest, q-LHP	9.900	7.440	1.077	0.923	132.993	3.421	12.833	2.534	77.185	1.991	15.411	0.350
		τ-latest, q-earliest	14.250	9.107	1.527	1.578	132.510	3.574	11.427	2.400	75.295	1.975	15.354	0.380
		τ-latest, q-latest	11.700	9.985	1.632	1.878	132.087	3.426	20.397	1.485	75.850	2.883	15.305	0.320
		τ-route, q-LHP	8.100	8.534	0.924	1.048	135.573	3.432	12.724	3.585	74.550	1.673	15.711	0.413
		τ-route, q-latest	7.400	8.068	1.042	0.898	136.989	3.011	22.970	1.615	73.765	1.808	15.875	0.371
		OFF	0.350	0.988	0.032	0.130	123.384	3.617	4.860	2.077	61.880	1.426	14.296	0.330
	tight	τ-earliest, q-earliest	12.200	16.938	2.372	3.017	141.820	3.805	8.029	3.336	64.295	3.578	16.608	0.526
		τ-earliest, q-latest	8.350	7.707	1.558	1.924	142.144	4.282	19.164	1.851	64.055	3.246	16.643	0.528
		τ-latest, q-LHP	7.050	8.587	1.121	1.442	133.109	5.952	11.178	4.118	76.335	2.057	15.575	0.261
		τ-latest, q-earliest	10.850	10.246	2.109	2.531	132.394	5.840	9.369	3.762	75.115	1.923	15.490	0.285
		τ-latest, q-latest	8.500	8.237	1.955	2.667	132.156	5.971	19.720	1.794	73.955	2.546	15.462	0.311
		τ-route, q-LHP	6.050	6.739	1.113	1.610	135.414	5.522	10.216	4.080	73.805	1.910	15.848	0.365
		τ-route, q-latest	5.050	6.700	1.021	1.536	136.518	4.566	21.582	1.895	73.285	2.089	15.980	0.347
		OFF	0.800	2.118	0.079	0.189	122.279	6.193	3.716	2.530	62.540	1.653	14.303	0.285

For τ-latest, q-LHP we use $\zeta = 1.2$ to convert the route performance values to the average time to serve an order as described in Section 7, and for τ-route, q-LHP $\zeta = 1.15$. This is only a naive transformation rule. Further research is needed regarding the driver performance estimation, especially since the waiting time, the route departure strategy, and the driver performance have an immanent cyclic dependency. The parameters used for the three-parameter inverse power law model to estimate the route performance $\phi(\lambda)$ given the load λ are tuned by means of least squares optimization to $k_l = 23.144, k_m = 3.951, \alpha = 0.174$, with a 25% estimated constant relative standard deviation. For the driver send home strategy, $\tilde{d}$ is set to 55 minutes.

10 Conclusions

We considered a purely dynamic and stochastic vehicle routing problem with delivery deadlines and shift flexibility arising from a real-world application. Orders arrive at an online store throughout the day, regarding which we have stochastic knowledge by means of hourly estimates. They have to be delivered within only a few hours by drivers deployed at a single depot. The goal is to reduce or evenly spread tardiness if not avoidable among the customers and to minimize travel and labor costs. Drivers may be sent home early or have their shifts extended to some degree to account for the uncertainty of the load.

Our proposed double-horizon approach is able to effectively address the dynamism and stochasticity of the problem. The large horizon planning with its simplified order-to-drivers assignment is able to derive meaningful target shift end times. These are exploited in the short horizon planning—the actual routing performed by ALNS—by augmenting its objective function, releasing additional shift resources in an informed way.

Another important aspect is to determine the route departure times, where neither the naive earliest nor the latest strategy suffices. We devised a more balanced strategy that estimates the expected route performance (average time per order in a route) depending on the current load and start routes earlier that are already close to the desired performance and later when the performance is worse.

The combination of both approaches leads to superior performance over the naive strategies or allows for trade-offs regarding substantially reduced travel and labor time versus a slight amount of more tardiness on artificially created instances for different load profiles (business day vs weekend) and shift plans (generous vs tight vs shortage).

Further research is needed to improve the estimation of the driver performance over the day, especially for real delivery areas in a city, also studying the impact of traffic. This is also a basis for the accuracy of the LHP. A difficulty lies in the cyclic dependency between the driver performance estimation and the route departure strategy, where we used a bootstrapping mechanism by fitting a function on idealized randomized instances, incorporating the driver waiting times by a constant factor over the whole day to pragmatically resolve this in a first step.

Acknowledgements We thank Daniel Obszelka for numerous fruitful discussions and the implementation and tuning of the destroy operators. We thank the anonymous reviewers for their valuable feedback resulting in several improvements of this work.

References

1. Azi, N., Gendreau, M., Potvin, J.Y.: An adaptive large neighborhood search for a vehicle routing problem with multiple routes. Computers and Operations Research **41**(1), 167–173 (2014)
2. Beardwood, J., Halton, J.H., Hammersley, J.M.: The shortest path through many points. In: Mathematical Proceedings of the Cambridge Philosophical Society, vol. 55, pp. 299–327. Cambridge University Press (1959)
3. Bent, R.W., Van Hentenryck, P.: Scenario-based planning for partially dynamic vehicle routing with stochastic customers. Operations Research **52**(6), 977–987 (2004)
4. Daganzo, C.F.: The distance traveled to visit n points with a maximum of c stops per vehicle: An analytic model and an application. Transportation science **18**(4), 331–350 (1984)
5. Figliozzi, M.A.: Planning approximations to the average length of vehicle routing problems with varying customer demands and routing constraints. Transportation Research Record **2089**(1), 1–8 (2008)
6. Gendreau, M., Guertin, F., Potvin, J.Y., Taillard, É.: Parallel tabu search for real-time vehicle routing and dispatching. Transportation Science **33**(4), 381–390 (2008)
7. Hvattum, L.M., Løkketangen, A., Laporte, G.: Solving a dynamic and stochastic vehicle routing problem with a sample scenario hedging heuristic. Transportation Science **40**(4), 421–438 (2006)
8. Mitrović-Minić, S., Krishnamurti, R., Laporte, G.: Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows. Transportation Research Part B: Methodological **38**(8), 669–685 (2004)
9. Pillac, V., Gendreau, M., Guéret, C., Medaglia, A.L.: A review of dynamic vehicle routing problems. European Journal of Operational Research **225**(1), 1–11 (2013)
10. Pisinger, D., Ropke, S.: A general heuristic for node routing problems. Computers & Operations Research **34**, 2403–2435 (2007)
11. Psaraftis, H.N., Wen, M., Kontovas, C.A.: Dynamic vehicle routing problems: Three decades and counting. Networks **67**(1), 3–31 (2016)
12. Ritzinger, U., Puchinger, J.: Hybrid metaheuristics for dynamic and stochastic vehicle routing. In: E.G. Talbi (ed.) Hybrid Metaheuristics, *SCI*, vol. 434, pp. 77–95. Springer (2013)
13. Ropke, S., Pisinger, D.: An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transportation Science **40**(4), 455–472 (2006)
14. Schilde, M., Doerner, K.F., Hartl, R.F.: Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports. Computers and Operations Research **38**(12), 1719–1730 (2011)
15. Shaw, P.: A new local search algorithm providing high quality solutions to vehicle routing problems. Tech. rep., APES Group, Dept. of Computer Science, University of Strathclyde, Glasgow, UK (1997)