Learning Surrogate Functions for the Short-Horizon Planning in Same-Day Delivery Problems*

Adrian Bracher, Nikolaus Frohner, and Günther R. Raidl

Institute of Logic and Computation, TU Wien
Favoritenstraße 9-11/192-01, 1040 Vienna, Austria
{nfrohner, raidl}@ac.tuwien.ac.at, adrian.bracher@live.at

Abstract. Same-day delivery problems are challenging stochastic vehicle routing problems, where dynamically arriving orders have to be delivered to customers within a short time while minimizing costs. In this work, we consider the short-horizon planning of a problem variant where every order has to be delivered with the goal to minimize delivery tardiness, travel times, and labor costs of the drivers involved. Stochastic information as spatial and temporal order distributions is available upfront. Since timely routing decisions have to be made over the planning horizon of a day, the well-known sampling approach from the literature for considering expected future orders is not suitable due to its high runtimes. To mitigate this, we suggest to use a surrogate function for route durations that predicts the future delivery duration of the orders belonging to a route at its planned starting time. This surrogate function is directly used in the online optimization replacing the myopic current route duration. The function is trained offline by data obtained from running full day-simulations, sampling and solving a number of scenarios for each route at each decision point in time. We consider three different models for the surrogate function and compare with a sampling approach on challenging real-world inspired artificial instances. Results indicate that the new approach can outperform the sampling approach by orders of magnitude regarding runtime while significantly reducing travel costs in most cases.

Keywords: Same-day delivery · Dynamic and stochastic vehicle routing · Sampling · Surrogate function optimization · Supervised learning.

1 Introduction

Short delivery times are essential when it comes to selling goods online, especially during the COVID-19 pandemic when many physical stores had to close temporarily. An increasing number of online retailers are offering same-day delivery

* This project is partially funded by the Doctoral Program "Vienna Graduate School on Computational Optimization", Austrian Science Foundation (FWF) Project No. W1260-N35.

to satisfy the demand for quickly available goods, further intensifying the need for cost and labor efficient dynamic vehicle routing. Same-day delivery problems [12] are stochastic and dynamic in nature and are a subcategory of vehicle routing problems. In this work a problem variant with additional constraints arising from practice is considered. Orders arrive dynamically over the day and are due only a short time after arrival. The orders have to be assigned to drivers and routes are generated with the goal to minimize delivery tardiness, travel times, and labor costs of the drivers involved. The fleet is homogeneous and the orders are served from a single depot. Each driver has a predefined shift, however the shift end times can be advanced or postponed to some extent to account for the uncertainty of the actual load.

This paper builds upon a double-horizon approach that was proposed in [3], which is further explained in Section 2.

However, we are unsatisfied with the existing short-horizon optimization, which we declare myopic, due to the following aspect: Routes that are optimal regarding all available orders sometimes have to start soon due to some orders, but also include one or more less urgent orders with a delivery deadline relatively far in the future. If these currently available orders with later deadlines introduce a significant travel overhead, it would frequently be wiser to postpone their delivery as they can likely be combined with future orders resulting in more efficient routes overall. Thus, it would be beneficial to split routes between urgent and less urgent orders. Routes can only be changed up to their departure, and possible future improvements for routes with a later starting time are not considered in the static, myopic optimization. The aim of this work is to present our adaptations to improve on this aspect.

The basic idea of our approach is to craft a function that discounts travel times based on the aforementioned observations, making separate routes with later starting times more attractive. We will refer to that function as surrogate, since it replaces the normal route duration in the objective function and also is used instead of a classical sampling approach, which is the de facto standard for stochastic considerations. This surrogate function is trained offline in a supervised learning fashion, reducing the computational effort in the online application in comparison to a sampling approach substantially. The necessary training data is generated by full-day simulations, in which we sample and solve 100 scenarios for each route at every decision point in time. Three different models for the surrogate are considered, a manually crafted exponential function, a linear regression, and a multi-layer perceptron.

In Section 2, an overview of related work is given and discussed. Section 3 defines and formalizes the problem at hand and aims to provide a better understanding with an illustrative example. Then, in Section 4, we explain our new approach in detail and describe the training data acquisition and training process in a step-by-step manner. Details on our test setup, and a comparison of our approaches with a sampling approach on real-world inspired artificial instances, can be found in Section 5. We observe that on our benchmark instances, the new approach reduces route travel costs by $\approx$6.1% in the median compared to the myopic optimization with similar tardiness. The sampling approach, in comparison, achieves a similar route duration reduction but requires a computation time that is larger by a factor of ≈540. We finally conclude in Section 6.

2 Related Work

For a review on dynamic and stochastic vehicle routing problems, see Ritzinger et al. [9]. Our underlying problem variant is introduced in [3] and derived from an online store with promised delivery durations of one or two hours. The problem is fully dynamic with a planning horizon of one day, where stochastic information regarding the hourly number and spatial distribution of orders is available upfront. A distinguishing feature is a flexibility of the shift ending times of the drivers, which is considered in the objective function together with route durations and a penalty for delivery deadline violations.

So far, the pillars of solving this problem are an adaptive large neighborhood search (ALNS) [10, 7, 1] for the repeated point-in-time optimization runs to obtain routes for currently available orders and a dual horizon approach inspired by Mitrović-Minić et al. [6]. At every decision epoch, a simplified assignment problem is solved for the larger horizon (i.e., the whole day) using expected values for the orders and driver performance. This allows an estimation of the required labor time which is subsequently fed back into the objective function used in the point-in-time optimization runs considering only short horizons. Near real-time decisions regarding planned assignments of orders to multiple trips of drivers, when to send drivers home, and when to start routes are then derived from the result of this optimization.

Due to the short delivery deadlines within few hours after customers place their orders, the problem falls into the class of *same-day delivery problems* (SDDP). In a recent notable work, Voccia et al. [12] present a SDDP variant with hard time windows where orders can also be delegated to a third-party, apart from delivering it with the in-house fleet. The number of orders to be delivered in-house is to be maximized and formulated as reward of a Markov decision process (MDP). A multiple scenario sampling approach (MSA) [2] is tailored to the problem, where at every decision epoch a multi-trip team orienteering problem with time windows is solved heuristically and a consensus solution is derived. This method increases the number of filled requests for some instance classes substantially compared to a simple delay strategy, where decisions are postponed to gather more information until an impact on the number of filled requests occurs. Still, a relevant drawback is that at every decision a couple of minutes computation time is required, making it unsuitable for our near real-time setting.

Despite the fact that we do not model our problem as an MDP explicitly, we perform implicit state transitions where actions (for each driver in the depot either wait, start an unalterable delivery route, or end shift) are derived from the heuristic solution. The goal of this work is to further adapt the objective function so that the implied actions lead to states with a higher expected reward in the near future.

Using the approximate dynamic programming paradigm [8], Ulmer et al. [11] solve a single-vehicle SDDP with preemptive depot returns using an Approximate Value Iteration (AVI) scheme where the value function is learned in an offline training phase over full-day simulations. Furthermore, a dynamic state space aggregation is used to create a lookup table facilitating near real-time online decision making.

Joe and Lau [4] build upon this approach for a dynamic vehicle routing problem with stochastic customers and different degrees of dynamism where re-routing decisions have to be performed on routing plans over the day. They replace the lookup table with a deep Q network employing value function approximation via temporal difference learning with experience replay. A heuristic search in the decision space is performed via simulated annealing. This approach is compared with AVI [11] and MSA [2], and the authors report reductions of the costs in the range of 10% for higher degrees of dynamism.

In a similar spirit, we approximate the value of states by predicting the future costs of orders that are in a currently planned route by means of parametric functions to be learned in an offline training phase with training data derived from multiple realizations in the short horizon—we consider the routes separately and do not roll-out until the end of the day, hence we make use of a vehicle-based and temporal decomposition. This learned function is then incorporated as surrogate in the objective function to be solved heuristically using our ALNS, resulting in more anticipatory online decision making.

3 Problem Definition and Formalization

In this Section we first give a formal description of the considered dynamic and stochastic vehicle routing problem and then show an illustrative example where myopic optimization in the short horizon planning leads to inefficient routes.

We follow the notation of the preceding article [3] and distinguish between three different 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). In this paper we will mostly focus on the dynamic variants.

3.1 Instance Specification

A DYN-DAY instance consists of many DYN-$\tilde{t}$ instances for increasing times $\tilde{t}$ that are solved iteratively over the whole day. A DYN-DAY instance contains n orders collectively denoted by V, each of which has a release time t_v^{rel} at which the order is ready to be delivered by a driver and a due time t_v^{due} at which the order should be delivered the latest, $v \in V$. Moreover, for each order an availability time t_v^{avail} is provided which corresponds to the time the customer places the order and tells us when we are allowed to consider the order in our planning. We further assume to have for dynamic instances a function $\omega(t_1, t_2)$ available that yields the expected numbers of orders within the time interval $[t_1, t_2]$ within any relevant business times. We also have an idea of the mean order duration, i.e., the mean active labor time by a driver to deliver an order, a good DYN-DAY solution in a particular application typically has and denote this by $\hat{\phi}$.

DYN-$\tilde{t}$ instances occur and are solved every time an order is released, i.e., at times $\left\{\tilde{t} \mid \exists v \in V \colon t = t_v^{\text{rel}}\right\}$.

Any problem instance also provides information about its relevant vehicles/drives, denoted by set U, with $m = |U|$, including each driver's planned shift time interval $[q_u^{\text{start}}, q_u^{\text{end}}]$ and earliest shift end $q_u^0 \in [q_u^{\text{start}}, q_u^{\text{end}}]$, $u \in U$. The drivers' shift ends are subject to flexibility and therefore also part of the decision process and objective function. Last but not least, order locations loc_v, $\forall v \in V$, expected travel times $\delta(v, v')$ from loc_v to $loc_{v'}$, where $v, v' \in V \cup \{0\}$, 0 denotes the warehouse, are given. The travel times include necessary delays like average stop time at the customers, average times for loading a vehicle at the warehouse and postprocessing times when returning to the warehouse. We also assume that the triangle inequality holds for δ.

3.2 Feasible Solutions

A candidate solution is a tuple $\langle R, \tau, q \rangle$ where

- $R = (R_u)_{u \in U}$ denotes the *ordered sequence of routes* $R_u = \{r_{u,1}, \dots, r_{u,\ell_u}\}$ to be performed by each vehicle $u \in U$, and each *route* $r \in R_u$ is an ordered sequence $r = \{v_0^r = 0, v_1^r, \dots, v_{l_r}^r, v_{l_r+1}^r = 0\}$ with $v_i^r \in V$, $i = 1, \dots, l_r$, 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_i^r of route r, $i = 1, \dots, l_r$, is delivered is

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

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 $\tau^{\min}(r) = \max_{i=1,\dots,l_r} t_{v_i^r}^{\text{rel}}$ be the earliest feasible starting time of a route r, which corresponds to the maximum release time of the orders served in the route. In our planning all routes $r \in R$ can be changed up to their respective departure time τ_r, after which the route is fixed. Furthermore, let $\tau^{\max}(r)$ be the latest starting time without violating any due time, i.e.,

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

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., $\tau_r \in [q^{\text{start}}_u, q^{\text{end}}_u]$, and not started before all 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., $q_u \geq \max(q^0_u, \sup_{r\in R_u}(\tau_r + d(r))),\ u \in U$.

3.3 Objective Function

The utmost goal is to minimize and balance any tardiness of deliveries. As secondary objectives, the route durations and the excess labor times are to be minimized. We model this via an objective function to be minimized consisting of a linear combination of a quadratic tardiness penalty term and linear cost terms for the secondary objectives:

$$f(\langle R, \tau, q\rangle) = \alpha \cdot \sum_{r\in R_u, u\in U} \sum_{i=1}^{l_r} \max(0, a(r,i) - t^{\text{due}}_{v^r_i})^2 + \gamma \cdot \sum_{u\in U}(q_u - q^0_u) + \sum_{r\in R_u, u\in U} d(r), \tag{4}$$

A tardiness penalty factor of $\alpha = 1000$ is chosen to approximate a lexicographic approach. The excess labor times cost factor γ is set to 4.

The objective values are not easily interpretable by humans. To give us another solution quality indicator we use the *mean order duration* $\phi(r) = \frac{d(r)}{l_r}$ for a route $r \in R$, measured in minutes, which can be understood as mean active labor time by a driver to deliver an order. Calculating the mean over all the routes in a solution yields $\bar{\phi}$, which is an important figure of merit of a whole solution.

3.4 Illustrative Example

To make the issue we address in this work clear, we present a simple example of a DYN-DAY instance, in which an optimal solution to a first DYN-$\tilde{t}$ instance leads to a situation so that an overall suboptimal solution for DYN-DAY is achieved.

Let us assume orders 1–6 become available at $\tilde{t}$=0 and we are thus considering DYN-0. Orders 1–5 are supposed to have the same due time 60 minutes later. The remaining order 6 is located far away from orders 1 to 5 and has a substantially

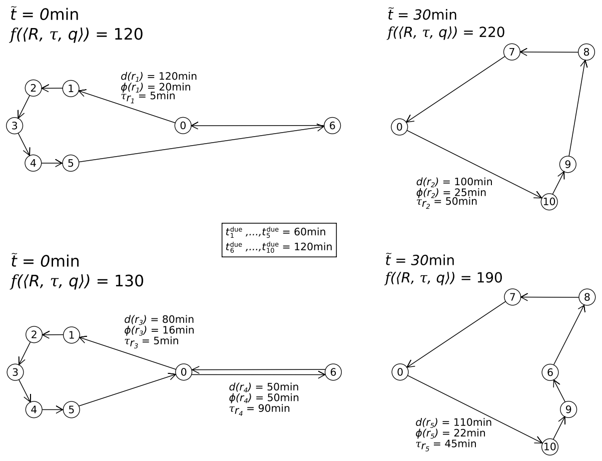

Fig. 1. Myopic solution (top) vs. optimal solution (bottom). Node 0 represents the warehouse, all other nodes orders.

later due time of 120 minutes. Considering only these orders an optimal solution to DYN-0 would be to pack all orders into one route, since only then the total route duration is minimal. This single route r_1 has to start at time $\tau_{r_1} = 5$ to avoid any tardiness. This solution is depicted in the top half of Fig. 1. The problem with this solution arises when half an hour later new orders 7–10 are placed with some delivery locations close to order 6, which itself, however, has been included in the already started first route. An optimal solution for DYN-30min will then be a route r_2 with the remaining orders 7–10 as also pictured in Fig. 1. The resulting total objective value, which in this case is equal to the sum of all route durations, is 220 minutes.

A better solution to this example can be seen in the second half of Figure 1. The important difference is that the first route from the previous solution is split into two, resulting in one route r_3 comprising orders 1–5, starting at time $\tau_{r_3} = 5$ and having a mean order duration better than the former single route, and one route r_4 containing only order 6. This latter route has a bad mean order duration of 50 minutes per order, but also a much later starting time of $\tau_{r_4} = 90$. Even though this results in a worse short-term objective value for DYN-0, this second route has now a lot of slack left and considering expected future orders can likely be improved later. In our example this happens when the new orders 7–10 become available in DYN-30, and order 6 can be delivered in one route r_5 together with the new orders. Overall the objective value and sum of all route durations for this solution is 190, which is an improvement of approximately 14% over the myopic solution.

In conclusion, when orders are expected in the near future, it makes sense to postpone to a certain degree the delivery of orders with due times farther in the future when they cannot be well integrated in soon-to-start routes.

4 Discounting Travel Times to Consider Expected Orders

As pointed out in Section 2, to at least partially avoid traps like the one sketched above arising from the myopic view of the DYN-$\tilde{t}$ instances, the standard method from literature is to sample scenarios into the near future by creating artificial orders from expected spatial and temporal distributions, to solve these scenarios, and then to derive a consensus solution [2, 12]. We propose the simpler approach of discounting durations of routes in the objective function of the DYN-$\tilde{t}$ instances in dependence of their starting times, the number of expected future orders, and further features. We make use of supervised learning to come up with a surrogate function for the route durations to move the computational effort into a one-time offline training phase. This function is then directly used in the optimization of a given DYN-$\tilde{t}$. We now describe our approach in detail.

In the definition of $f(\langle R, \tau, q\rangle)$ in (4) on page 6 we replace the route duration $d(r)$ of each route $r \in R_u,\ u \in U$ with a *discounted duration* $d'(r)$ acting as a surrogate for the future delivery time of the orders belonging to r. We define the following aims to guide us to a sensible discounting function.

– Routes that are already efficient, i.e., have a low mean order duration $\phi(r)$, should not be modified.
– The discounting of the duration should in general be stronger when more orders are expected in the near future. On the contrary, we should not reduce $d(r)$ if there are no further orders expected until route r should start.
– Routes that are inefficient and combine orders that are due soon with orders that have significantly more time left should be avoided in particular.
– In conclusion, the discounted route duration $d'(r)$ should approximate the *expected total time it will take to perform the deliveries of that route in the future*, taking into account expected new orders and assuming optimal routing decisions also in the future.

A current route that will be started soon cannot be expected to be improved much as not many new orders are expected. This includes routes with small slack $\max(0, \tau^{\max}(r) - \tilde{t})$ but also any other case in which the route is started soon, e.g., due to an earliest starting time strategy. In contrast, larger improvements are likely for any route that is planned to be started much later and which is not yet efficient, particularly if many orders are expected in the near future, more precisely in the time interval from the current time $\tilde{t}$ to the route's planned starting time τ_r. Thus, this duration is an important parameter of the discounting.

Another important parameter is the expected number of arriving new orders until the start of the route, i.e., $\omega(\tilde{t}, \tau_r)$. Moreover, the estimated mean order duration of a good DYN-DAY solution $\hat{\phi}$ is also important for the following consideration. A route r to be started at some distant time τ_r and whose mean order duration $\phi(r)$ is worse than $\hat{\phi}$ can be expected to be adapted and combined with future orders so that the average times for delivering the orders in r approaches $\hat{\phi}$.

Formally, we model this by the *discounted route duration function*

$$d'(r) = \begin{cases} g_{\Theta}(d(r), l_r, \omega(\tilde{t}, \tau_r), \hat{\phi}, \ldots) & \text{if } \tau_r > \tilde{t} \wedge \phi(r) > \hat{\phi} \\ d(r) & \text{else.} \end{cases} \tag{5}$$

where function g_{Θ} represents a machine learning model with trainable parameters Θ and input features that include at least $d(r),\ l_r, \omega(\tilde{t}, \tau_r)$ and $\hat{\phi}$. This model is supposed to yield reduced durations within $[\hat{\phi} \cdot l_r, d(r)]$ for routes that are not started immediately ($\tau_r > \tilde{t}$) and where the current mean order duration $\phi(r)$ is worse than $\hat{\phi}$. In Section 4.2 we will consider three different approaches for realizing g_{Θ}, which are an exponential function, a linear regression, and a multilayer perceptron.

An aspect of this approximation that deserves mentioning is that multiple routes of the current solution may be scheduled at overlapping times in the future and may compete for new orders. This may slow down improvement of inefficient routes but may also create new possibilities for more efficient combinations. As we do not see any meaningful and efficient way to consider this aspect and also conjecture that the benefits and disadvantages of concurrent routes in conjunction with the route improvement potential may outweigh each other at least to a certain degree, we do not explore this further here. Moreover, the actual impact may be partially mitigated by suitably tuning Θ.

4.1 Obtaining Training Data

To obtain training date for our route duration discounting models, we apply the following sampling-based approach on a set of representative historic or artificial DYN-DAY training instances.

1. We consider a DYN-DAY instance and iteratively solve the implied DYN-$\tilde{t}$ instances in the classical way without any route distance discounting. For each obtained DYN-$\tilde{t}$ solution, we apply a decomposition approach, in which we consider each route independently by the following steps.
2. Each route r to be started not immediately, i.e., at some time $\tau_r > \tilde{t}$, and for which $\phi(r) > \hat{\phi}$, we create n_{sample} scenarios, with n_{sample} being a strategy parameter. Each scenario consists of the original orders of route r and $n_{\text{orders}} \sim \mathcal{P}(\omega(\tilde{t}, \tau_r))$ additional artificial orders, where n_{orders} is a random number always sampled anew from the Poisson distribution $\mathcal{P}(\omega(\tilde{t}, \tau_r))$ with mean $\omega(\tilde{t}, \tau_r)$. The motivation here is that the arrival of orders can be seen as a Poisson process. Each artificial order is assigned a randomly sampled geographical location from a set of sufficient size representing the delivery area, a random availability time in $(\tilde{t}, \tau_r]$, and a due time that corresponds to the availability time plus the maximum delivery duration promised to the customers. Each scenario created this way is then solved as an independent OFF instance.
3. In each obtained scenario solution we consider each original (i.e., not sampled) order and take its route's mean order duration. The sum of these times

over all original orders is then said to be the scenario's total delivery duration for the original orders of route r. Ultimately we average these total delivery durations over all scenarios to obtain the target duration $\hat{d}(r)$ which we want to approximate by our discounted route duration $d'(r)$.

4. We store the original route r together with $d(r)$, $\tilde{t}$, τ_r, $\omega(\tilde{t}, \tau_r)$, $\hat{\phi}$ and the obtained $\hat{d}(r)$ for training and continue by processing all further routes in the same way.

4.2 Models for the Discounting

As introduced in Eq. (5), function $g_\Theta(d(r), l_r, \omega(\tilde{t}, \tau_r), \hat{\phi}, \ldots)$ is a trainable model that yields the discounted route duration when $d(r) > \hat{\phi} \cdot l_r$. For training this model we apply the mean squared error (MSE) in respect to the training targets, i.e., $\hat{d}(r)$, as loss function. We investigate here three alternative models presented in the following.

Exponential Function.

$$g_\rho^{\exp}(d(r), l_r, \omega(\tilde{t}, \tau_r), \hat{\phi}) = d(r) - (d(r) - \hat{\phi} \cdot l_r) \cdot (1 - e^{-\rho \cdot \omega(\tilde{t}, \tau_r)}) \tag{6}$$

This function was manually crafted based on the previously explained considerations that the mean order delivery time of orders in a current route with a distant starting time can be expected to improve up to a certain amount. The expected maximum improvement is assumed to be equal to $d(r) - \hat{\phi} \cdot l_r$. However, actual improvement can only occur with additional orders in the interval $(\tilde{t}, \tau_r]$. This is expressed by the last term in the function, where parameter ρ controls the speed of approaching $\hat{\phi} \cdot l_r$ in dependence of the number of expected upcoming new orders $\omega(\tilde{t}, \tau_r)$ until the route's starting time τ_r in an exponential manner. The parameter that needs to be learned here is just $\Theta = \rho$, and we apply grid search to find a value minimizing the MSE.

Linear Regression. Our second approach is a linear combination of a larger set of manually selected features, i.e., linear regression, with the trainable parameters vector Θ being the respective regression coefficients. We initially consider the following features in addition to a constant bias.

1. The basic features $d(r)$, l_r, $\omega(\tilde{t}, \tau_r)$ and $\hat{\phi}$ as in the exponential function.
2. The relative starting time of the route $\tau_r - \tilde{t}$.
3. The difference $\phi(r) - \hat{\phi}$, i.e., how far off the route's mean delivery duration is from the assumed target value $\hat{\phi}$.
4. The *variance of the geographic locations of the orders for each route*, denoted by $\mathrm{var}(r)$; the farther apart the delivery locations are, the more likely it seems that a new order fits nicely in between two existing orders.
5. The square and the logarithm of each of the above features to also accommodate nonlinear dependencies in a simple form.

To avoid the inclusion of features that do not significantly improve the prediction and reduce the danger of overfitting, we started off with just the basic features and iteratively added a feature from the remaining pool that reduced the MSE the most. This process of selecting features was continued until the MSE did not change by more than one percent. 5-fold cross validation was used in this feature selection process to reduce the risk of overfitting. Ultimately, we came up with the feature vector $(d(r),\ l_r,\ \omega(\tilde{t},\tau_r),\ \log(\omega(\tilde{t},\tau_r)),\ \hat{\phi},\ \phi(r)-\hat{\phi},\ (\phi(r)-\hat{\phi})^2,\ (\tau_r-\tilde{t})^2,\ \log(\tau_r-\tilde{t}),\ \mathrm{var}(r),\ \mathrm{var}(r)^2)$ used in all further experiments.

Multilayer Perceptron. Our third model for discounting travel durations is a multilayer perceptron (MLP). It is fully connected with two hidden layers and ReLU activation functions in all layers, and Adam [5] is used as optimizer. The considered pool of features was the same as in the linear regression, and the same selection process was performed leading to the feature vector $(d(r),\ \log(d(r)),\ l_r,\ \hat{\phi},\ \tau_r-\tilde{t},\ \phi(r)-\hat{\phi})$ used in all following experiments. Note in particular that here the variance of the orders' geographic locations did not show a significant contribution and therefore was not included. Further details on the network configuration and training will be provided below in the experimental results.

5 Computational Study

All algorithms were implemented in Python 3.8. Training and evaluation of the regressors was performed with `scikit-learn` version 0.23.1. All tests were conducted on Intel Xeon E5-2640 2.40 GHz processors in single-threaded mode and a memory limit of 4 GB.

In all tests a driver is sent home as early as possible, i.e., after the driver's last so far planned route or at the earliest shift end, to minimize labor cost. Planned routes always start at the latest possible departure time that does not increase the costs for labor time and tardiness to utilize the full slack for possible improvement. The three different discounting models are compared with results using the myopic optimization as done in [3] and the sampling approach with consensus function. The ALNS, which is the fundamental optimization method for all mentioned approaches, stops after 100 non-improving iterations, and we refer to [3] for all further details concerning its operators and configuration.

5.1 Instances

We consider artificial DYN-DAY instances that are inspired by real-world instances of an online retailer. We consider steady, linearly rising, and falling load patterns over 11 hours, where the average load over the day is either 10, 20, 30, or 40 arriving orders per hour. Orders are due in one hour with 60% probability and with 40% in two hours. The order locations are uniformly distributed in the unit square. Travel times between orders are determined by the Euclidean distance multiplied by 50 minutes, additional constant six minutes stop times at the customers, and small loading and postprocessing times from and to the warehouse. The warehouse location is randomly chosen from $\{0.25, 0.75\}^2$ inspired by the slight off-center location of the real-world situation. Since we focus on the route duration costs, sufficiently many drivers are available all the time to ensure zero or very little tardiness. We generated 240 instances in total, 20 for each of the 12 instance classes and perform a 50/50 training and test split. $\hat{\phi}$ is provided for each class. All instances were made available on GitHub[1].

5.2 Training of the Discounted Route Duration Models

Following the training and test data generation as described in Section 4.1, we end up with a 60% batch of 33790 training samples and a 40% batch of 22527 test samples to train and evaluate an estimator for $\hat{d}(r)$.

We train the learnable parameter ρ in the exponential model (6) by means of a grid search. The result can be seen in Figure 2, which displays how the MSE changes depending on ρ. Moreover, the instance's $\hat{\phi}$ is reduced by 20%, which was empirically determined to produce better results in previous experiments. The single global optimum for ρ is 0.091, at which the test MSE is 154 909 and 154 265 for the training batch.

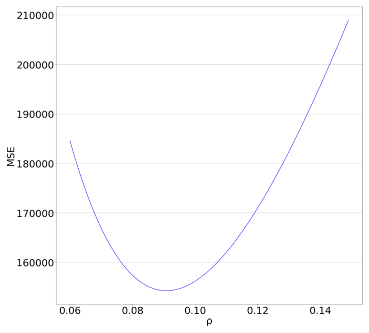

Fig. 2. Exponential model: MSE of predicted values $g_\Theta(r)$ with respect to labels $\hat{d}(r)$, i.e., the loss over ρ.

In case of the linear regression with the finally selected features as laid out in Section 4.2, MSEs of 144 785 and 143 329 were achieved on the training and test portions of the data, respectively.

Concerning the MLP, preliminary tests suggested that two hidden layers with 50 nodes each seem to be a reasonable choice, which we used further on. The learning rate that is used for training is a constant 0.001. To avoid overfitting we utilize early stopping, for which 300 iterations without improvement of a 10% validation set is the stopping criterion. The resulting training and test MSEs are 80 032 and 79 219 respectively, slightly less than half of the error of the exponential model.

Concerning the MSEs, we can conclude that the linear regression performs slightly better than the exponential model, but the MLP is clearly superior. As we considered separate training and test sets and the respective MSEs lie close together for all three models, we conclude that overfitting seems to be no issue for all three models.

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

5.3 Full-day Simulation Results

The myopic short-horizon optimization serves as a baseline to quantify the improvement that is achieved. Furthermore, the three route duration discounting approaches are compared to a sampling approach with consensus function as the de facto standard for considering stochastic aspects. This approach creates for each DYN-$\tilde{t}$ instance 100 scenarios by augmenting the original instance with randomly sampled orders. These sampled orders are generated in the same manner as already explained in Section 4.1, except that the time interval $[\tilde{t}, \tilde{t} + 1h]$ is used instead of the slack of the route, i.e., samples are generated for up to one hour into the future. These scenarios are then solved with the myopic short-term optimization utilizing ALNS. Then, all sampled orders are removed from each scenario solution. Finally, a consensus solution is derived from the scenario solutions in a way that was inspired by [12]. The selection is done by counting identical scenario solutions and choosing the most frequent solution as consensus solution. We define identical in this context as two solutions that assign identical routes to the same drivers in the same sequence. Analogous to that identical routes are defined as routes that contain the same orders in the same sequence.

We use the original objective function $f(\langle R, \tau, q\rangle)$ as defined in (4) as the primary measure of success for comparing results, but also aim to gain a more in-depth understanding of the different approaches by observing the total duration of all routes in a solution, the total excess labor time of a solution, the mean order duration over the whole solution $\bar{\phi}$ and the running time on the specified test setup. Tardiness is not presented in this Section, because it is negligibly small for all instance classes and approaches alike, which was one of our aims when generating the test instances as explained in Section 5.1.

In Table 1 the median of the mentioned measures of success are compared for all instance classes and the median of the relative changes to results of the myopic approach is displayed for the most important measures as well. As the sampling approach did not terminate within a time limit of 700 hours per full-day instance for average loads of 30 and higher, we only obtained results up to an average load of 20 for it. In Figure 3 boxplots of $f(\langle R, \tau, q\rangle)$ are drawn over instance classes grouped by the average load as well as the load pattern.

As expected, all approaches that consider possible future orders outperform the myopic optimization, up to 8% in the median. We observe that the exponential approach outperformed the other approaches for average loads of 30 and 40. Furthermore, a positive correlation between the average load and the relative improvement over the myopic short-term optimization can be seen. Falling load solutions have higher $f(\langle R, \tau, q\rangle)$ in general, but the differences in relative improvement over the myopic optimization among load patterns is rather small, with steady load having a slight edge over falling and rising load.

Considering that the MLP has the smallest training MSE, it is unexpected to observe that some solutions are worse than the ones that utilize the exponential model. We suspect that the cause for this is attributed to the way in which the training data is generated. More specifically, we intentionally decided to restrict the training data generation to routes in final DYN-$\tilde{t}$ solutions obtained from

Table 1. The three discounting approaches, myopic optimization, and sampling applied to ten benchmark instances for each combination of average load and a falling, rising, or steady load as the day progresses.

Load	Pattern	Approach	$f(\langle R, \tau, q\rangle)$		Trav. time [h]		Labor	ϕ		Runtime [min]
Avg.			Median	Change	Median	Change	Median	Median	Change	Median
10	Falling	Myopic	4057.258	0.00%	67.197	0.00%	136.5	36.125	0.00%	4
		Exponential	3921.867	-2.92%	64.297	-3.14%	424.5	34.920	-3.14%	5
		Linear Regression	**3851.000**	-4.28%	**64.069**	**-4.32%**	**141.0**	35.170	**-4.33%**	8
		MLP	3912.190	**-4.96%**	64.349	-3.92%	147.9	**34.404**	-3.92%	11
		Classical Sampling	3928.320	-4.20%	64.634	-3.80%	201.4	35.157	-3.79%	2950
	Rising	Myopic	3816.300	0.00%	63.413	0.00%	172.5	35.485	0.00%	4
		Exponential	3695.408	-3.49%	61.564	-3.32%	**37.0**	33.920	-3.31%	5
		Linear Regression	**3662.175**	-3.57%	**61.036**	-3.59%	117.5	34.505	-3.58%	8
		MLP	3701.557	**-3.89%**	61.693	**-3.79%**	61.6	**33.976**	**-3.81%**	12
		Classical Sampling	3749.525	-1.14%	62.464	-0.85%	45.5	34.845	-0.85%	3347
	Steady	Myopic	3984.472	0.00%	66.040	0.00%	**0.0**	35.475	0.00%	5
		Exponential	3891.581	-4.31%	64.456	-5.01%	**0.0**	34.000	-5.00%	5
		Linear Regression	3938.683	-4.97%	65.258	-5.63%	135.0	33.795	-5.63%	7
		MLP	3845.711	-5.54%	63.349	-5.48%	**0.0**	33.064	-5.48%	8
		Classical Sampling	**3769.011**	**-8.56%**	**62.771**	**-8.41%**	3.5	**32.618**	**-8.41%**	2543
20	Falling	Myopic	7142.592	0.00%	118.993	0.00%	77.0	32.015	0.00%	24
		Exponential	6802.900	-5.90%	112.105	-5.87%	19.0	**30.025**	-5.87%	32
		Linear Regression	6823.300	-5.40%	112.755	-5.58%	25.0	30.265	-5.57%	49
		MLP	6884.071	-5.10%	112.935	-5.10%	**1.0**	30.360	-5.10%	51
		Classical Sampling	**6693.054**	**-6.32%**	**111.639**	**-6.18%**	4.0	30.127	**-6.18%**	30372
	Rising	Myopic	7027.803	0.00%	116.848	0.00%	380.0	32.365	0.00%	23
		Exponential	6482.008	-6.51%	107.817	**-6.53%**	177.0	**30.965**	-6.53%	32
		Linear Regression	**6419.892**	**-6.62%**	**106.955**	-6.52%	136.0	31.055	**-6.54%**	40
		MLP	6432.858	-6.54%	107.044	-6.22%	124.2	31.065	-6.20%	62
		Classical Sampling	6641.478	-3.79%	110.440	-3.72%	**102.0**	32.138	-3.72%	33728
	Steady	Myopic	7101.992	0.00%	117.963	0.00%	252.5	32.690	0.00%	26
		Exponential	6765.575	**-6.77%**	112.431	**-6.60%**	137.5	31.465	**-6.59%**	28
		Linear Regression	6830.842	-4.02%	113.435	-3.86%	127.5	31.585	-3.85%	37
		MLP	**6721.294**	-5.68%	**111.578**	-5.53%	95.1	31.519	-5.54%	52
		Classical Sampling	6735.598	-5.93%	112.078	-5.60%	**94.0**	**31.060**	-5.60%	25646
30	Falling	Myopic	10432.136	0.00%	172.496	0.00%	**487.5**	31.150	0.00%	77
		Exponential	**9657.147**	**-7.07%**	**159.930**	**-7.01%**	681.0	29.030	**-7.00%**	95
		Linear Regression	9721.894	-5.92%	160.721	-5.55%	713.0	**29.025**	-5.55%	135
		MLP	9689.704	-5.95%	160.690	-5.44%	714.7	29.072	-5.45%	176
	Rising	Myopic	10313.425	0.00%	170.299	0.00%	1308.5	31.055	0.00%	76
		Exponential	9687.325	-6.66%	160.255	**-6.66%**	802.5	**28.690**	**-6.68%**	99
		Linear Regression	9766.358	-6.26%	162.100	-6.14%	**385.5**	29.150	-6.14%	134
		MLP	**9599.087**	**-6.68%**	**159.226**	-6.06%	569.4	29.196	-6.04%	146
	Steady	Myopic	10378.903	0.00%	171.450	0.00%	867.5	31.460	0.00%	82
		Exponential	**9633.436**	**-6.94%**	**159.578**	-6.64%	629.0	**29.225**	**-6.62%**	76
		Linear Regression	9772.233	-5.61%	161.868	-5.32%	**305.5**	29.895	-5.34%	112
		MLP	9802.089	-4.79%	162.408	-4.82%	675.5	29.394	-4.83%	156
40	Falling	Myopic	12632.717	0.00%	209.611	0.00%	508.5	29.530	0.00%	149
		Exponential	**11713.483**	**-7.83%**	**194.497**	**-7.90%**	247.0	**27.420**	**-7.88%**	193
		Linear Regression	11970.428	-6.76%	198.876	-6.56%	**241.5**	27.465	-6.57%	295
		MLP	12031.577	-6.41%	199.944	-6.38%	414.8	27.629	-6.36%	425
	Rising	Myopic	12837.467	0.00%	212.832	0.00%	969.5	30.170	0.00%	195
		Exponential	**12005.597**	**-6.65%**	**199.567**	**-6.58%**	494.5	**27.675**	**-6.57%**	234
		Linear Regression	12238.508	-6.36%	203.612	-6.21%	**408.5**	28.025	-6.20%	311
		MLP	12042.505	-6.57%	199.835	-6.22%	615.2	27.862	-6.21%	332
	Steady	Myopic	12635.717	0.00%	209.439	0.00%	883.0	29.335	0.00%	178
		Exponential	**11715.214**	**-8.04%**	**194.479**	**-8.16%**	540.5	**27.000**	**-8.16%**	170
		Linear Regression	11798.203	-6.49%	196.503	-6.46%	**407.0**	27.560	-6.48%	259
		MLP	11836.576	-7.42%	196.776	-7.45%	535.3	27.341	-7.46%	309

the ALNS. The reasoning behind that decision is that we want to avoid an overwhelmingly large number of routes that are very bad, to derive finer tuned models for better routes, which usually end up in the solution. This is especially bad for the linear regression and the MLP that are more closely fitted to the training data, whereas the exponential function benefits in this regard from its simplicity and robustness.

6 Conclusions and Future Work

We considered a same-day delivery problem in which dynamically arriving orders have to be delivered within a short time span while minimizing travel times, la-

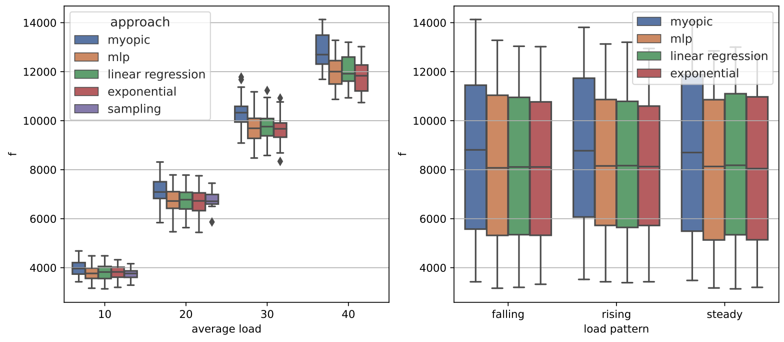

Fig. 3. Solution quality $f(\langle R, \tau, q \rangle)$ over average load and load pattern. The sampling approach is not included in the load pattern graphic due to missing data for average loads greater than 20.

bor costs, and tardiness. We focused on incorporating stochastic knowledge into the objective function of the point-in-time optimization runs, realized by an ALNS, by discounting route durations in dependence of diverse features. The most important features are the number of orders that can be expected up to the latest time the route would need to be started and the route's mean delivery duration, but several other factors were also considered and partly showed significant benefits.

Overall, our experiments clearly indicated that this approach is able to alleviate to a substantial degree the weaknesses of a myopic optimization, in particular in higher load situations. Of the three route duration discounting models the exponential function performs the best, reducing the travel time as well as the total objective by ≈6.1% on average over all instance classes. The more flexible neural net, in contrast, performed significantly weaker. We conjectured that the reason for this at the first glance surprising observation is the bias we have in the training data. The simpler exponential function seems to be more robust concerning candidate routes with properties that do not appear so frequently in the routes determined by the ALNS when generating training data. Moreover, the independent consideration of the routes is another source of potential errors. In our experiments, the exponential discounting even outperformed the sampling approach regarding solution quality in most cases and cuts down on runtime by several orders of magnitude.

Further work should consider alternative ways of generating training data to possibly reduce the bias. For example, intermediate solutions of the ALNS may occasionally also be used for data generation. Bootstrapping $\hat{\phi}$ from previous non-myopic runs could improve the accuracy of the parameter and lead to further improvement. Moreover, the variability of this mean order duration over the day due to varying load and traffic should be considered. Also, further tests with real-world inspired spatial order distributions (e.g., clustered instances) and load patterns could be helpful to evaluate practical aspects of the discounting models.

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. Bent, R.W., Van Hentenryck, P.: Scenario-based planning for partially dynamic vehicle routing with stochastic customers. Operations Research **52**(6), 977–987 (2004)
3. Frohner, N., Raidl, G.R.: A double-horizon approach to a purely dynamic and stochastic vehicle routing problem with delivery deadlines and shift flexibility. In: Causmaecker, P.D., et al. (eds.) Proceedings of the 13th International Conference on the Practice and Theory of Automated Timetabling - PATAT 2021: Volume I. Bruges, Belgium (2020)
4. Joe, W., Lau, H.C.: Deep reinforcement learning approach to solve dynamic vehicle routing problem with stochastic customers. In: Proceedings of the International Conference on Automated Planning and Scheduling. vol. 30, pp. 394–402 (2020)
5. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
6. 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)
7. Pisinger, D., Ropke, S.: A general heuristic for node routing problems. Computers & Operations Research **34**, 2403–2435 (2007)
8. Powell, W.B.: Approximate Dynamic Programming: Solving the curses of dimensionality, vol. 703. John Wiley & Sons (2007)
9. Ritzinger, U., Puchinger, J., Hartl, R.F.: A survey on dynamic and stochastic vehicle routing problems. International Journal of Production Research **54**(1), 215–231 (2016)
10. 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)
11. Ulmer, M.W., Thomas, B.W., Mattfeld, D.C.: Preemptive depot returns for dynamic same-day delivery. EURO Journal on Transportation and Logistics **8**(4), 327–361 (2019)
12. Voccia, S.A., Campbell, A.M., Thomas, B.W.: The same-day delivery problem for online purchases. Transportation Science **53**(1), 167–184 (2019)