



Distance approximation to support customer selection in vehicle routing problems

Fabian Akkerman¹ · Martijn Mes¹

Accepted: 14 March 2022
© The Author(s) 2022

Abstract

Estimating the solution value of transportation problems can be useful to assign customers to days for multi-period vehicle routing problems, or to make customer selection decisions very fast (e.g., within an online environment). In this paper, we apply several regression methods to predict the total distance of the traveling salesman problem (TSP) and vehicle routing problem (VRP). We show that distance can be estimated fairly accurate using simple regression models and only a limited number of features. Besides using features found in the scientific literature, we also introduce new classes of spatial features. The model is validated on a fictional case with different spatial instances considering both a backordering and lost sales configuration, and on a realistic case that involves dynamic waste collection in the city of Amsterdam, The Netherlands. For the fictional case, we show differences in performance per instance type and configuration, and we show that our model can save up to 25.3% in distance compared with a heuristic approximation. For the waste collection case, we introduce a cost function that combines the travel distance and service level, and show that our model can reduce distances up to 17% compared to a well-known heuristic approximation while maintaining the same service level. Furthermore, we show the benefits of using approximations for combining offline learning with online or frequent optimization.

Keywords Distance approximation · Traveling salesman · Vehicle routing · Customer selection · Inventory routing · Waste collection

1 Introduction

With the increasing size and complexity of logistics problems, there is an increasing demand for fast solution methods to solve vehicle routing problems (VRPs). Estimating the (optimal) solution value has received more and more attention in operations research studies in recent years. Approximating the solution value of VRPs, i.e., the total distance, can help solving multi-period VRPs and other logistics problems where customer selection or assignment is

✉ Fabian Akkerman
f.r.akkerman@utwente.nl

Martijn Mes
m.r.k.mes@utwente.nl

¹ High Tech Business and Entrepreneurship Department, University of Twente, Enschede, The Netherlands

necessary. When vehicle capacities are insufficient to serve all customers within a given time period, or when customers need to be assigned a delivery day for a multi-period problem, we need to select a subset of customers to serve. Examples of customer selection in practice are the collection of oil from oil fields with a limited fleet of oil trucks (Duhamel et al., 2009), a large-scale manufacturer buy-back campaign causing peaks in demand for collecting disposed products from dealers at variable buy-back prices (Aras et al., 2011), the need for a fast selection of feasible delivery time slots to offer to e-retail grocery consumers (Agatz et al., 2011), and the selection of dynamically arriving customers for parcel pickup services with limitations on working hours (Ulmer et al., 2018). These, and other problems often need to be solved in stages to cope with the multi-period structure, or require a fast response in online situations. Our distance approximation models can help to decompose multi-period problems into single-period sub-problems, or make customer selection decisions in an environment with high demand on computational times.

In this paper, we develop an approximation method that utilizes regression models to approximate the costs within transportation problems related to distance. We consider both the traveling salesman problem (TSP), creating the shortest route for a single vehicle visiting a given set of customers, and the vehicle routing problem (VRP), considering a fleet of vehicles with capacity restrictions. Both the TSP and the VRP are NP-hard combinatorial optimization problems, which means that realistic instances can typically only be solved heuristically (Dror et al., 1994). We validate our model by applying it to two case studies: a fictional case study with different (spatial) settings, and a real case on dynamic waste collection in Amsterdam, The Netherlands. Fast approximation techniques can be advantageous for both cases; as the problem instances are relatively large and demand is stochastic, we have to consider a longer-term planning horizon. The stylized case study is introduced to study the effects of different spatial patterns on the performance of our proposed model. Our distance approximation model supports the customer acceptance decision for the upcoming delivery day. Furthermore, we show how our model can be applied to a backordering case (i.e., customers can be postponed indefinitely) as well as to a lost sales case (i.e., customer sales are lost when not fulfilled within a time interval after their arrival). To avoid excessive computational time in online or frequent decision making, we propose a combination of offline learning (training the approximation model) and operational decision making utilizing the approximation model.

The remainder of this paper is structured as follows. In Sect. 2, we introduce the relevant scientific literature on combinations of online and offline methods, customer selection, distance approximation, and waste collection problems. Furthermore, we describe how we extend the current literature and highlight our contributions. In Sect. 3, we introduce our approximation model and discuss the combination of offline learning and online optimization. In Sect. 4, we describe the stylized customer selection case and the waste collection case. In Sect. 5, we validate and illustrate our model using the two case studies. We close with conclusions and future research directions in Sect. 6.

2 Literature

We briefly review the literature on methods that reduce the computational demands for solving the VRP. We first discuss the use of offline methods that support online or operational decision making, e.g., assigning customers to clusters or time slots before making a routing decision. Next, we treat the literature about customer selection and discuss related works about TSP and VRP distance approximations. Finally, we treat the relevant literature about modelling the planning of waste collection related to our case study of waste collection in Amsterdam.

VRP research increasingly considers real-life, dynamic environments, which typically involve stochastic demands, stochastic travel times, and other disturbances (Braekers et al., 2016). This means that VRP models need to come up with robust plans that can handle changing environments. Furthermore, VRP complexity increases by considering multiple periods, multiple depots, larger (heterogenous) fleets, and larger customers bases. As opposed to exact solutions, approximation methods are typically more robust and generally better able to solve large multi-period problems, hence they are more often applied in real-life situations (Caceres-Cruz et al., 2014). With longer planning horizons, and therefore larger problem sizes, the need for faster solution methods increases. There are numerous options for limiting computational complexity, both for exact and approximate methods. The decision space can be reduced by, e.g., disregarding indisputably bad decisions, prioritizing customers, or restricting the decision space to the cheapest options (Gromicho et al., 2012). Other methods focus on improving fast obtained heuristic solutions using metaheuristics, or split the heuristic solution in smaller sub-parts that can be solved to optimality using exact methods (Lalla-Ruiz & Voß, 2020). The multi-period VRP considered in Bard and Nananukul (2009), is split in two stages: first, a linear program is used that assigns delivery quantities to customers by maximizing an estimated term for customer value, next, a single-day VRP is solved exactly. Approximating the unknown solution value, prior to solving, can help to reduce the decision space by excluding potentially weak solutions or unattractive problem instances. To keep computational effort low, a model can be split in an offline and online phase. Training the model on historic data can be done offline, while the application of the model is considered an online phase, since costs are incurred during the decision making process (Powell & Ryzhov, 2013).

Several authors use offline methods to improve online decisions. In Ulmer et al. (2019), approximate dynamic programming (ADP), also known as value-based reinforcement learning, is applied to the uncapacitated single-vehicle routing problem with stochastic service requests. Their approach has an offline value function approximation (VFA) component, which determines the value of a state using a heuristic and simulation, as further described in Ulmer et al. (2018). The state is defined by (i) the time of arrival to the current vehicle location, and (ii) the time budget, which is defined as the time left until the duration limit. The online routing decisions are then made using the already known VFA. They conclude that the geographical spread of customers is a good predictor for the success of an approximation. The approach in Novoa and Storer (2009) is similar in the usage of ADP. They, however, define the state to be the current vehicle location, remaining vehicle capacity, and the demand yet to be delivered. Both approaches enable fast, online decision making by shifting the computational effort to an offline stage.

Large scale, complex, multi-period VRPs drive the need for heuristic approximation. But besides the approximations used to speed up the computations, some VRP variants more explicitly require some form of customer selection or prioritization. Examples of these problem variants are the vehicle routing problem with profits (VRPP), inventory routing problem (IRP), and problems where customers arrive dynamically during the day (Ulmer et al., 2018). Approximation approaches are needed for customer selection problems since it takes too much computational effort to evaluate all possible customer subsets. The number of possible subsets, with subset size r , from a set of customers with size n , equals $n!/r!(n-r)!$. Selecting a subset of customers to serve is relevant when capacity is insufficient to serve all customers. The Vehicle Routing Problem with Profits (VRPP) is a problem variant where the total collected profits minus travel costs is maximized, while adhering to time or capacity constraints (El-Hajj et al., 2020). For the VRPP it is allowed to leave cost-unattractive locations unvisited. In Vidal et al. (2016), a solution for the VRPP is proposed that first assigns all

customers to vehicles and solves the respective TSPs. Next, a SELECT algorithm is applied to each route, activating a subset of the customers while fulfilling the resource constraints and maximizing profits. El-Hajj et al. (2020) propose a particle swarm optimization algorithm (PSO) to maximize the number of served customers for the multi-period VRPP. Their algorithm extracts the best routes from a given particle, and improves the extracted routes with selective neighborhood operators, e.g., swapping or moving served customers, or removing served customers and inserting unserved customers (El-Hajj et al., 2020). Tricoire et al. (2010) consider a multi-period variant of the VRPP, the multi-period orienteering problem with time windows. Similar to El-Hajj et al. (2020), they use neighborhood operators with a metaheuristic (variable neighborhood search) to move, select and deselect customers, e.g., a customer is moved to a different tour and, hence, a different day (Tricoire et al., 2010). These methods all decrease computational demand by partitioning the VRP and solving the smaller problems with exact or heuristic methods. The way the larger problem is partitioned has major influence on the global solution quality. A method that approximates the objective value of a sub-problem can improve the quality of partitioning and therefore improve the global solution. Partitioning VRPs by means of customer prioritization is also considered for inventory routing problems (IRP). The IRP combines the fields of inventory management (when to serve customers and how much to deliver to each customer) and routing (how to route the vehicles along the selected customers) (Coelho et al., 2014). In Roldán et al. (2016), several rule-based policies are tested on IRP instances where vehicle capacity is insufficient to replenish all customers fully. Their policies prioritize the customers with the highest demand or lowest inventory level, respectively.

Formulas for approximating TSP and VRP distance have received ample attention in the scientific literature. Beardwood et al. (1959) proved that the shortest TSP distance, serving N customers in a bounded plane of area size A , is almost always asymptotically proportional to \sqrt{AN} for large N . Further development of the TSP distance formulas were done by, amongst others, Christofides and Eilon (1969), Chien (1992), and Hindle and Worthington (2004). For the capacitated VRP (CVRP), a well-known formula is the Daganzo-approximation. The formula is a fairly accurate approximation of the CVRP distance (Robusté et al., 1990) and is calculated as follows:

$$\text{CVRP distance} \approx \left[0.9 + \frac{kN}{C^2} \right] \cdot \sqrt{AN}, \quad (1)$$

where k is an area shape constant, and C is the maximum number of customers a vehicle can serve. Later, their formula was improved by considering the shape of the area (Robusté et al., 2004). In addition, Figliozzi (2008) tested several formulas for the VRP as well as for the VRP with time windows (VRPTW), in different spatial settings. Their models show high performance on Solomon instances (Solomon, 1987), the best performing approximation is calculated with:

$$\text{VRP distance} \approx k_l \frac{N-M}{N} \sqrt{AN} + k_b \sqrt{\frac{A}{N}} + k_m M, \quad (2)$$

where the parameters k_l , k_b , and k_m are determined with linear regression, and M is the number of available vehicles.

Aside from approximation formulas, research has been conducted on the use of machine learning methods to approximate the value of a vehicle routing decision. Kwon et al. (1995) use linear regression and neural networks in combination with several spatial features (e.g., average customer-depot distance, area of the service region, and area of the smallest rectangle that covers all customers) for predicting TSP tour length. Their models can estimate TSP

distance fairly accurate. In Arnold and Sörensen (2019), the characteristics of a VRP solution are described by several features. Using classification algorithms (decision-trees, random forests, and support vector machines), they distinguish good and bad solutions. Their research shows that a good heuristic can be further improved by guiding the search process using classification data, e.g., by removing edges that are unlikely to appear in a good solution. The research in Nicola et al. (2019) is focused on the prediction of travel distances using linear regression with several customer-oriented features, like geographic information and demand. They show that the approximation of distance is accurate for the TSP and VRP, especially for Solomon instances with clustered customers (Solomon, 1987).

The planning of waste collection is gaining a lot of attention in the scientific literature (Beliën et al., 2014). The main focus is on the collection of household waste. The collection from larger containers is typically modelled as an adapted VRP, also called the Waste Collection Vehicle Routing Problem (WCVRP) (Beliën et al., 2014). WCVRPs have as objective to find optimal routes for collecting waste from a set of containers. Collection vehicles leave the depot empty, collect waste, and unload waste at a disposal facility when the route is completed or when the vehicle capacity has been reached. At the end of the day, the vehicles return to the depot (Benjamin & Beasley, 2010). The WCVRP requires the set of containers to be emptied to be known upfront. A distinctive feature of WCVRPs is the dynamicity, which entails the influence of today's decisions on the next-day decision space (Baita et al., 1998). We distinguish the following options for including dynamicity: (i) run the model for a long planning horizon, (ii) solve a periodic VRP (PVRP), which concerns multi-period problems like in Archetti et al. (2017), and (iii) model the planning of waste collection as an inventory routing problem (IRP). The IRP is a medium-term problem, in contrast to the short-term character of the regular VRP or WCVRP (Archetti et al., 2017). The classification in Heijnen (2019) shows that most IRP literature uses a one-to-many topology, i.e., instances with a single depot serving many customers. Some authors extend the problem with satellite facilities, which function as additional depots, effectively increasing vehicle capacity (Bard et al., 1998). Research on the planning of waste collection is done for both single-period models and multi-period models. However, since the long-term planning approach has positive effects on long-term outcomes (Moin & Salhi, 2007), contemporary research tends to treat multi-period models (Heijnen, 2019). In Mourgaya and Vanderbeck (2007), a mixed integer programming method is proposed for household waste collection in rural areas. Their model integrates the routing decisions with waste collection site selection in a multi-period setting with unknown demand. Their computational results illustrate the complexity of the waste collection problem, since small real world instances already require excessive computational effort. The application of the IRP to a medical waste collection problem is considered in Taslimi et al. (2020). They consider the planning of collection of hazardous medical waste. Their design concerns a weekly inventory routing schedule with as goal to minimize transportation costs and risks related to storing hazardous materials. The proposed decomposition based heuristic divides the problem into single-period problems, after which an integer program is solved for the routing (Taslimi et al., 2020). Another application of the IRP to a waste collection problem can be found in Mes (2012), who studies the added value of a dynamic planning methodology compared to a cyclic emptying schedule. In a follow-up study, Mes et al. (2014) propose a dynamic collection policy with tunable parameters to adjust the policy to changing environments. The parameters are tuned using optimal learning techniques in a simulation optimization approach.

In a preparatory study, we tested various features for the waste collection case (Akkerman et al., 2020). In this paper, we build further on this by introducing a new class of features for the VRP and analysing various approximations for both the waste collection case as well

as for generic multi-period VRPs. Our contribution to the scientific literature is threefold. First, we develop a solution methodology that supports customer selection decisions in routing problems, as well as the decomposition of multi-period vehicle routing problems. Second, we introduce several new spatial features to better estimate the characteristics of VRP solutions. Third, we illustrate our solution methodology using a stylized customer selection case study in different spatial settings and a waste collection case.

3 Distance approximation model

In this section, we subsequently introduce the approximation model for the TSP in Sect. 3.1 and later extend it for the VRP in Sect. 3.2. With respect to the models, we make a clear separation between generic elements and case-specific elements, enabling the model to be applied to a variety of problems. Furthermore, for the generic model, we distinguish between the TSP and VRP. Since VRPs include multiple vehicles, the basic TSP model needs to be expanded to consider vehicle capacity and expected demand. Various features are proposed and evaluated using (i) linear regression, (ii) random forests regression, (iii) lightGBM, and (iv) multi-layer perceptron regression, i.e., neural networks. We describe how the data is generated in Sect. 3.3 and we show the application of automatic feature selection techniques for both linear regression and random forests in Sect. 3.4. In Sect. 3.5, we show how we apply an automatic hyperparameter tuning method to obtain the best settings for random forests and neural networks. Next, we evaluate the predictive performance of our approximation models for both the TSP and VRP in Sect. 3.6. We end with Sect. 3.7, where we introduce our framework for improving the approximation by combining online optimization and offline learning. This adaptive learning framework summarizes the contribution of this research.

3.1 Model for the traveling salesman problem

For the approximation of a TSP route, we only consider spatial features and disregard the customer demand data. The features are summarized in Table 1 and further explained below.

F1 is the number of customers visited by a vehicle. F2 is the area size of the smallest possible rectangle that fits all visited locations, including the depot. F3 is the perimeter of this rectangle. Alternatively to the enclosing rectangle, the area and perimeter can be calculated by taking the convex hull around all locations, including the depot (F4, F5). F6 and F7 are the width and height of the enclosing rectangle, respectively. F8–F13 are several distance related features. F8 is the average distance between customers and F9 is the average distance between customers and the depot. The customer centroid (centre of mass) is found by averaging all latitudes and longitudes. The rectangle centroid is the the point where the two diagonals of the enclosing rectangle intersect. The customer centroid and rectangle centroid are used for F10–F13 to calculate the (average) distance from the depot or all customers to the respective points.

The angle related features F14–F16 express the dispersion of the customers by taking the variance of several different bearings between customers and either the depot (F14), customer centroid (F15), or rectangle centroid (F16). The bearing $\beta_{a,b}$ between point a and b , is the angle between the line connecting the two points and the north-south line of the earth, and can be calculated with (3), (4) and (5).

Table 1 Summary of features for the TSP

Feature ID	Feature or feature type	Source
F1	Number of customers	Rasku et al. (2016), Arnold and Sørensen (2019), Nicola et al. (2019)
F2, F3	Enclosing rectangle (area and perimeter)	Kwon et al. (1995), Rasku et al. (2016)
F4, F5	Convex Hull (area and perimeter)	Rasku et al. (2016)
F6, F7	Enclosing rectangle (width, height)	Rasku et al. (2016)
F8	Avg. distance between customers	Rasku et al. (2016), Arnold and Sørensen (2019), Nicola et al. (2019)
F9	Avg. distance depot to customers	Kwon et al. (1995), Rasku et al. (2016), Arnold and Sørensen (2019), Nicola et al. (2019)
F10, F11	Distance depot to rectangle or customer centroid	Rasku et al. (2016), Arnold and Sørensen (2019), Nicola et al. (2019)
F12, F13	Avg. distance customers to rectangle or customer centroid	Rasku et al. (2016), Arnold and Sørensen (2019), Nicola et al. (2019)
F14–F16	Angle related	Arnold and Sørensen (2019)
F17–F22	Geographical variance	Nicola et al. (2019)
F23–F28	Radius related	–
F29–F36	Rectangular partitioning	–

$$x = \cos(lat_b) \cdot \sin(\Delta(lon_a, lon_b)) \quad (3)$$

$$y = \cos(lat_a) \cdot \sin(lat_b) - \sin(lat_a) \cdot \cos(lat_b) \cdot \cos(\Delta(lon_a, lon_b)) \quad (4)$$

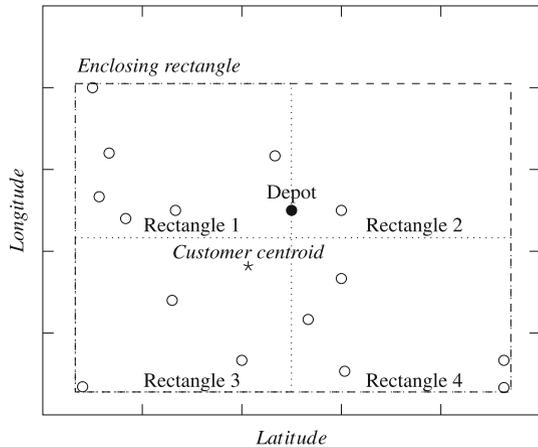
$$\beta_{a,b} = \arctan(x, y) \quad (5)$$

F14–F16 are based on latitude and longitude but can be converted to a Cartesian system by substituting the north-south line by one of the Cartesian axes. F17–F22 express geographical variance and dispersion by means of variance in customer latitude and longitude (F17), the variance of customer latitude multiplied with longitude (F18), and the variance of the distance from customers to either the depot (F19), customer centroid (F20), or rectangle centroid (F21), and the variance of distances between all customers (F22) in a route.

In addition to these features based on literature, we introduce two new types of features. First, F23–F28 count the number of customers within a certain radius from the depot, customer centroid, or rectangle centroid, respectively. We use two different radius sizes: 0.5 M and 0.75 M , where M is the distance between the respective circle centrepoint and the furthest away customer. These radius features have similar descriptive power as other already described features but might be more convenient to calculate. Second, F29–F33 are rectangular partitioning features. We split the smallest possible rectangle that can be fitted around all locations into several equally sized smaller rectangles. An illustrative example of a 2×2 rectangular partitioning structure is depicted in Fig. 1.

In our experiments, we test two different rectangular partitioning structures, namely a 10×10 structure and a 15×15 structure. Several features can be extracted from the rectangular partitioning structure: the distance between the depot and the centroid of the rectangle with the most customers (F29), e.g., Rectangle 1 in Fig. 1; the average distance from the depot to the centroid of all activated rectangles (F30), i.e., rectangles that contain customers; the average distance between activated rectangle centroids (F31); and the average distance

Fig. 1 Illustrative instance depicting a 2×2 rectangular partitioning structure inside the rectangle that encloses all visited locations



between customers grouped inside a rectangle (F32). F33–F36 are similar features, but calculated for the 15×15 rectangle setting. The rectangular partitioning features can capture the extent of concentration of customers at geographical locations.

3.2 Model extensions for the vehicle routing problem

For the VRP, we consider spatial data as well as demand data. The addition of demand data is imperative since the VRP involves multiple vehicle routes and capacitated vehicles. The additional features considered for the VRP are shown in Table 2. Here, F37–F42 describe the VRP instance considering the demand and vehicle capacity. The variance of customer demand d_i (F39) is calculated with all n customers included in the VRP:

$$S_d^2 = \frac{\sum_{i=0}^I (d_i - \bar{d})^2}{n}. \quad (6)$$

F41 is similar to F40, but rounds up to the nearest integer. F43 is a feature for which we count the rectangles (10×10 setting) in which the demand is higher than the average rectangle demand. As alternative to the rectangular partitioning features F29–F36, we propose a

Table 2 Summary of additional features for the VRP

Feature ID	Feature	Source
F37	Total demand per instance	Nicola et al. (2019)
F38	Avg demand per instance	Nicola et al. (2019)
F39	Variance of demand per instance	Nicola et al. (2019)
F40	$\frac{\text{Total demand per instance}}{\text{Vehicle capacity}}$	Nicola et al. (2019)
F41	Minimum required vehicles	Rasku et al. (2016)
F42	$\frac{\text{Maximum customer demand in an instance}}{\text{Vehicle capacity}}$	Rasku et al. (2016)
F43	Count high demand rectangles	–
F44–F48	Seed clustering related	–

different method to group customers in a route into subsets over which features are calculated. We first initialize C empty customer clusters, where $C = \left\lceil \frac{\text{Total demand}}{\text{Vehicle capacity}} \right\rceil$. Next, a clustering algorithm is applied, roughly based on Fisher and Jaikumar (1981). First, C seeds are chosen by selecting customers that are furthest from the depot and the other selected seeds. Next, an assignment algorithm groups customers to seeds based on the smallest distance from the seed to the respective customer. We calculate the average distance between customers in clusters (F44), between centroids of customer clusters (F45), and between all customers in a cluster and either the cluster centroid (F46) or the depot (F47). Also, we calculate the average distance from the depot to the the furthest away customer in a cluster (F48).

3.3 Data generation

We use a simulation model we developed to mimic the waste collection planning of Amsterdam, The Netherlands, to generate data for training our approximation models. More detailed information about the waste collection case, and the corresponding simulation model, can be found in Sect. 4.2. Household containers are selected and emptied on the respective days of the simulation. Each day in the simulation, a container selection algorithm selects a subset of all containers that will be emptied during the current day. The vehicle routes are constructed using a cluster-first-route-second approach and improved with a 2-opt metaheuristic. A simplification, in comparison with the actual case as described in Sect. 4.2, is that we only focus on planned routes and ignore possible disruptions during the day, caused by, e.g., additional trips to the depot because of higher fill levels than expected. Also, we only consider routes starting and ending at the central depot, without considering the satellite facilities as used in the waste collection case study. Hence, this choice of generating the vehicle routes as training data does not affect the generic applicability. For creating training data for the TSP, we select the routes of a single vehicle for each day. For the TSP training data, we store the locations, including the location of the depot, and the distance per vehicle. For obtaining the VRP training data, the TSP data is extended with demand data (fill levels) and next, the vehicle routing data is aggregated per day, i.e., the aforementioned TSP data is combined if the TSP routes were planned on the same day. The obtained TSP and VRP data, with each 15,000 entries, is split into a training set and a validation set, in a 80%/20% ratio. Features are standardized before training.

3.4 Feature selection

Feature selection is performed for several reasons. First, it indicates the individual importance of the features for the regression models. Second, features might be correlated or suffer from multicollinearity, which potentially can distort some models. In addition, a model can be overfitted because there are too many features relative to the available data. Finally, the computational time needed to calculate the feature values needs to be as low as possible for the approximation to be fast enough (Rasku et al., 2016). Therefore, even though some models are robust to noise from bad features (Hastie et al., 2009), it can be valuable to evaluate the feature importance.

We employ two different methods for feature selection. The first method is called Elastic Net Regularization (ENR) (Zou & Hastie, 2005). ENR combines two linear regression methods: Lasso regression with L_1 penalization and Ridge regression with L_2 penalization. By combining the two methods, the advantages of both methods can be exploited, and

the limitations reduced. Lasso regression shrinks large feature coefficients and can be an effective tool for automatic feature selection. However, the Lasso fails to select grouped features, i.e., features that suffer from multicollinearity (Zou & Hastie, 2005). Ridge regression, however, does recognize grouped features but does not do automatic feature selection. ENR successfully combines these two methods. First, the assumptions for using linear regression need to be reviewed; we observe that the residuals are by estimation normally distributed and homoscedastic, i.e., we can safely assume linear regression is a valid method for our data. Note that features were standardized before fitting as this is necessary for coefficient shrinkage methods.

The second method, called Boruta-Shap (BShap), is used for the tree-based methods. BShap employs an iterative procedure of copying features and randomizing them to remove correlation with the target. These new “shadow” features are compared with the regular features, allowing for calculating statistical significant feature importance scores. We use a variant of the algorithm that employs Shapley-values as internal importance measure, as using this permutation-based statistic aids the process of finding global feature importance (Keany, 2020; Orlenko & Moore, 2021).

For more details on both selection methods, we refer to Zou and Hastie (2005) and Kursu and Rudnicki (2010). Finally, all features are selected for the neural networks regressor, since neural networks are better able to learn complex relationships and weigh the importance of features.

3.5 Hyperparameter tuning

Hyperparameter tuning is an important procedure for machine learning models. For both tree-based methods and neural networks, there are several settings that can influence the performance and the chance of overfitting. We tune the hyperparameters on the training set using Bayesian optimization with the Scikit-Learn and Scikit-Optimize Python libraries (Pedregosa et al., 2011; Head et al., 2021). Bayesian optimization iteratively samples different values of hyperparameters within a wide interval. The algorithm is efficient since it incorporates prior belief about the best hyperparameter values from previous iterations to direct new sampling and trade-off exploration and exploitation (Brochu et al., 2010). As scoring criteria we use R^2 , which is the variance in the data set that can be explained by the model. The R^2 of selected models is measured using a 5-fold cross validation procedure.

Table 3 shows the best settings for TSP and VRP data, found by Bayesian optimization for the tree-based methods, random forests regressor (RFR) and lightGBM, respectively. LightGBM is a gradient boosting method that is often more accurate and efficient compared to standard tree-based methods, like RFR (Ke et al., 2017). We set the number of trees to 200 for all models, striking a balance between performance and computational effort. Next, we let the trees grow to their full depth and tune the maximum number of features to consider when splitting a tree (RFR), the maximum number of bins that features will be bucketed

Table 3 Hyperparameters found by Bayesian optimization with 5-fold cross validation for the tree-based methods

Hyperparameter	TSP	VRP
Max. features for splitting (RFR)	30	42
Max. bins (lightGBM)	139	160
Learning rate (lightGBM)	0.0584	0.0341

Table 4 Hyperparameters found by Bayesian optimization with 5-fold cross validation for the neural networks

Hyperparameter	TSP	VRP
Initial Learning rate	0.038	0.0024
Batch size	78	187

in (lightGBM), and the learning rate (lightGBM). We apply bootstrapping and out-of-bag samples to the tree-based methods to reduce the chance of overfitting (Hastie et al., 2009).

We determine the neural network architecture with the following procedure. On a separate data set, we determine the number of hidden layers and nodes in such a way that the neural network is almost perfectly fitted to the training set of size 17,000, i.e., the R^2 is close to 1.0 and the error is close to 0. Next, we reduce the complexity of the architecture until we do no longer see an overfit, i.e., the performance decreases on the training set and increases on the validation set. After performing this procedure, we find an architecture of 3 hidden layers, with 128, 64, and 32 hidden neurons, respectively. We use this architecture for both the TSP and VRP models, since it seems to suffice for both instance types. For both the TSP and VRP model, the Adam weight optimization solver is used, as proposed by Kingma and Ba (2015). Adam is a stochastic gradient solver that works well on large data sets (Kingma & Ba, 2015). Furthermore, we use ReLU as the activation function and adaptive weight updates, i.e., the learning rate is constant, equal to the initial learning rate, but is divided by 5 when two successive epochs fail to decrease training loss or increase the validation score by at least 0.0001. Next, we tune the initial learning rate and batch size with Bayesian optimization. The settings found by the Bayesian optimization algorithm for neural networks are summarized in Table 4.

3.6 Model performance

We compare models using five different statistics: adjusted R^2 , relative mean absolute error (rMAE), relative root mean squared error (rRMSE), mean percentage error (MPE), and mean absolute percentage error (MAPE). The adjusted R^2 is adjusted for the number of features in the model. The measures rMAE and rRMSE provide an indication of the quality of the approximation. The regular MAE indicates the average magnitude of errors without considering direction. The regular RMSE penalizes large errors more than the MAE, and therefore the RMSE is useful for identifying prediction outliers. Both MAE and RMSE are made relative to the mean of the observed values. The MPE is the average of all percentage errors and indicates whether the prediction underestimates or overestimates the actual distance. The MAPE is the average percentage of absolute error. With respect to the latter two measures, we note that MPE tends to balance out the errors and MAPE is biased when the actual values are small.

$$\text{rMAE} = \frac{\frac{1}{N} \sum_{i=1}^N |\text{Predicted}_i - \text{Actual}_i|}{\text{Actual}} \times 100\% \quad (7)$$

$$\text{rRMSE} = \frac{\sqrt{\frac{\sum_{i=1}^N (\text{Predicted}_i - \text{Actual}_i)^2}{N}}}{\text{Actual}} \times 100\% \quad (8)$$

$$\text{MPE} = \frac{1}{N} \sum_{i=0}^N \frac{\text{Predicted}_i - \text{Actual}_i}{\text{Actual}_i} \times 100\% \quad (9)$$

Table 5 Comparison of approximation formulas, literature based features, and new features on TSP and VRP data, using linear regression

Statistics	TSP			VRP		
	\sqrt{AN}	Literature features	Literature and new features	Daganzo	Literature features	Literature and new features
(Adjusted) R^2 (out-of-sample)	0.467	0.910	0.923	0.657	0.844	0.871
rMAE (out-of-sample)	50.21%	8.91%	8.46%	15.37%	9.25%	8.08%
rRMSE (out-of-sample)	57.27%	9.87%	9.5%	18.89%	12.56%	10.67%
MPE (out-of-sample)	-5.67%	-3.56%	-3.24%	2.34%	-3.23%	-3.03%
MAPE (out-of-sample)	10.47%	7.68%	7.17%	13.57%	9.17%	8.97%

$$\text{MAPE} = \frac{1}{N} \sum_{i=0}^N \frac{|Predicted_i - Actual_i|}{Actual_i} \times 100\% \quad (10)$$

Before we show the performance of the various regression models, we first study the predictive performance of closed-form approximation formulas for the TSP (Beardwood et al., 1959) and VRP (Robusté et al., 1990). We compare these closed-form formulas with (i) the features as suggested by the literature, and (ii) the features suggested by the literature extended with our proposed features, including the radius, rectangle partitioning, and seed clustering features. We compare using our generated data and employ linear regression as the predictor. Table 5 shows the results of this comparison on the validation set (out-of-sample).

For the TSP data, it appears that the formula by Beardwood et al. (1959) is not able to predict TSP distance accurately, reporting an R^2 of 0.467. Even though this formula was proven to be asymptotically proportional to the TSP distance, it cannot capture TSP distance for these instances, possibly due to the complex, non-convex area with a limited number of customers. The difference in performance between the features proposed in the literature and our extensions is small, with an R^2 of 0.910 and 0.923, respectively, which is a slight improvement. The same small difference is apparent for the other error statistics. We conclude that for the TSP, our proposed features result in a small improvement of predictive performance. For the VRP, we observe that the Daganzo-formula is able to capture the VRP distance for these instances, but still the R^2 is low with 0.657. The other error statistics confirm this. The comparison between the literature-based features and our proposed extensions shows that the addition of our proposed features has a modest but positive effect on predictive performance. The adjusted R^2 rises from 0.844 to 0.871 and the various error statistics show a similar effect when adding our proposed features. In the remainder of this section, we compare various regression models on the complete feature data set, while employing the proposed feature selection methods.

Table 6 shows the performance of the four models on the TSP training and validation data set. We observe that the predictive performance is high for all models. On the training set (in-sample), our models perform similarly to the models presented in Hindle and Worthington (2004) and slightly worse compared with the regression models in Nicola et al. (2019), who both report performance on stylized instances. The negative MPE indicates that all our models tend to slightly underestimate the actual TSP distance.

Comparing the models with the validation set performance, Elastic Net Regularization (ENR) seems to perform the worst, with a rMAE of 7% of the average distance in the validation set, and an R^2 of 0.938. Random forests regression (RFR) performs slightly better

Table 6 Model performance for the TSP

Statistics	ENR	RFR	lightGBM	NN
Number of Features	26	31	32	36
Adjusted R^2 (in-sample)	0.945	0.995	0.995	0.986
rMAE (in-sample)	5.63%	1.75%	1.62%	2.02%
rRMSE (in-sample)	6.8%	2.38%	2.03%	2.15%
MPE (in-sample)	-0.99%	-0.21%	-0.15%	-0.23%
MAPE (in-sample)	4.27%	1.79%	1.65%	1.53%
Adjusted R^2 (out-of-sample)	0.938	0.965	0.97	0.948
rMAE (out-of-sample)	7%	4.91%	4.72%	4.99%
rRMSE (out-of-sample)	7.86%	6.58%	6.42%	6.62%
MPE (out-of-sample)	-1.89%	-0.56%	-0.38%	-0.42%
MAPE (out-of-sample)	8.49%	5.03%	4.99%	5.12%

Table 7 Model performance for the VRP

Statistics	ENR	RFR	lightGBM	NN
Number of Features	38	32	36	48
Adjusted R^2 (in-sample)	0.896	0.99	0.99	0.97
rMAE (in-sample)	4.34%	1.71%	1.61%	1.76%
rRMSE (in-sample)	8.23%	2.82%	2.11%	2.11%
MPE (in-sample)	-1.42%	-0.23%	-0.11%	-0.21%
MAPE (in-sample)	5.33%	2.56%	1.63%	2.21%
Adjusted R^2 (out-of-sample)	0.876	0.944	0.961	0.941
rMAE (out-of-sample)	4.7%	2.87%	2.12%	2.71%
rRMSE (out-of-sample)	10.21%	3.4%	3.1%	4.25%
MPE (out-of-sample)	-2.94%	-0.34%	-0.29%	-0.44%
MAPE (out-of-sample)	6.6%	3.07%	2.98%	3.99%

than neural networks (NN), with a rMAE of 4.91% and R^2 of 0.965. The relative RMSE is 6.58%, which indicates that there are not many large outliers. The best performing model is lightGBM, with an adjusted R^2 of 0.97, rMAE of 4.72% and rRMSE of 6.42%. For all models, the MPE on the validation set is between -0.38% and -1.89%, while the MAPE lies between 4.99% and 8.49%. Note that, even for the in-sample case, NN does not outperform the tree-based methods because overfitting is prevented using the L2 regularization penalty.

ENR eliminates 10 features and BShap eliminates 5 features for RFR and 4 features for lightGBM. ENR removes 2 out of the 3 radius features (F23–F28), only the feature representing the proximity to the depot is kept. Some seemingly good features are removed, possibly because of redundancy. All rectangular partitioning features (F29–F36) are in the model. BShap makes a different selection: it removes 4 of the 8 rectangular partitioning features, with the only one remaining being the average number of customers in an activated rectangle.

Table 7 shows the performance of the four models on the VRP data set. Compared with the TSP model, the performance of ENR drops significantly. We observe that, both on the

training set and the validation set, the R^2 decreases and the MPE and MAPE increases. The higher rRMSE indicates that there are some large outliers that heavily influence the performance. Nevertheless, the rMAE is reasonably good, with 4.34% on the training set and 4.7% on the validation set. The regression models that can better handle nonlinear relationships outperform ENR and show similar predictive performance on the TSP and VRP data. LightGBM outperforms the random forests regression and neural networks. Our approximation model for the VRP outperforms the models presented by Figliozzi (2008) and Nicola et al. (2019), when compared with their reported performance on random clustered instances, that closely resemble our data.

ENR removes several seemingly redundant features but keeps all demand-related features (F37–F42). BShap now removes more features compared to the TSP model. Both feature selection methods keep all seed clustering features (F44–F48) in the model. For the TSP and VRP model, we observe that the highest Shapley-values are given to the following features: number of customers (F1), enclosing rectangle perimeter (F3), convex hull area (F4), enclosing rectangle width and height (F4,F7), average distance between locations (F8), the multiplied variance of customer latitudes and longitudes (F17), the distance from the centroid of all activated rectangles to the depot (F30), and the average distance from the depot to the furthest away customer in a seed cluster (F48).

3.7 Adaptive learning framework for improving approximations

The main advantage of learning models, as opposed to heuristic methods, is that they can be retrained and adapt to changing circumstances. The method of training a model (offline), using the approximation to optimize decisions (online), and retraining a model again is shown in Fig. 2. For our case studies, we do not consider online optimization, since decisions are

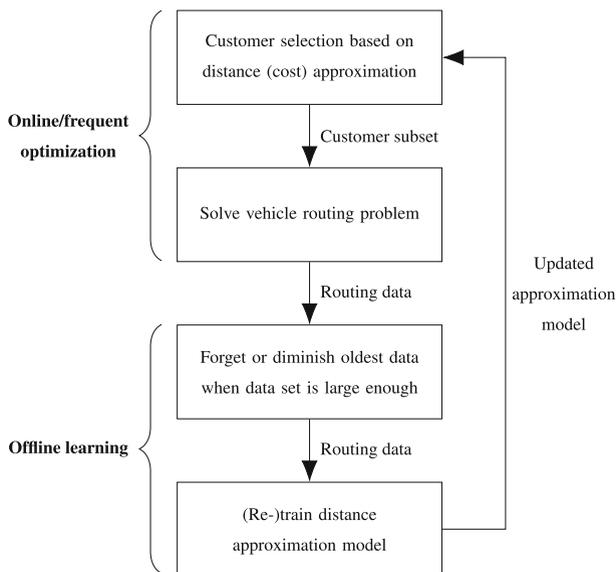


Fig. 2 Adaptive learning feedback loop with a customer selection phase, route construction phase, and model training phase

only made at the start of the day, not during the day. Nevertheless, this procedure can also be applied to frequent optimization cases like ours. The framework can be applied to cases where the environment changes and the approximation model needs to be updated regularly, e.g., when customer demand changes or the geographic area of operations changes. Alternatively, the framework can be used to improve the approximation of a stable environment by obtaining more data.

In the first phase, customer selection decisions are made based on a distance (cost) approximation model. Next, in the second phase, a routing schedule is constructed that delivers all selected customers. As soon as the data set collected during the iterations of the feedback loop is large enough, the oldest data can be forgotten or given less importance in phase three. The routing realization data is used in the fourth phase to train or retrain an approximation model. Finally, the new approximation model is used in the first phase again. This process, as depicted in Fig. 2, generalizes the process used for the stylized customer selection case and the waste collection case, and summarizes the contribution of this paper. In Sect. 4, we will show how this framework can be applied to our case studies.

4 Case studies

In this section, we introduce our case studies. First, we describe the stylized customer selection case in Sect. 4.1. Next, we describe our waste collection case study in Sect. 4.2. Finally, in Sect. 4.3, we explain how we adapt our generic approximation model to cope with a combined cost term of distance and service level.

4.1 Settings for the stylized customer selection case

We use a fictional case to test our proposed regression model for the vehicle routing problem and study the performance of customer selection for the VRP. We decompose the daily decision process in two stages. First, customers are selected based on the lowest expected increase in routing costs as described by our approximation. Next, a vehicle routing problem is solved for the subset of customers that can be delivered with the limited vehicle capacity. Before the assignment decision at the start of day t , a random number of customers $c_t \in \mathcal{C}_t$ arrives, drawn from a discrete uniform distribution, $|\mathcal{C}_t| \sim U[a, b]$. So, all customers are known before creating the routes for the current day, and a subset of these customers needs to be selected in order to adhere to capacity constraints and minimize routing distance. Before the first execution day ($t = 0$), there are already $|C_0|$ initial customers present in the system. Each customer has a demand d_c , which is drawn from a discrete uniform distribution, $d_c \sim U[a_d, b_d]$. All customers are served from the single depot, with a homogeneous fleet of K vehicles, each vehicle having a capacity of Q . The values for K and Q and the parameters for d_c are chosen in such a way that approximately half of the daily customers can be served. The other half needs to be postponed to the next day. The parameters of the discrete uniform distribution that determine the number of daily arriving customers $|\mathcal{C}_t|$ are chosen in such a way that the system is in a steady state, i.e., the daily number of customers fluctuates around a stable mean. The starting state of the system is chosen in such a way that half of the fleet capacity is already reserved for existing customers C_0 , before new customers arrive on the first day. Since our proposed regression models first need data to be trained, we start by collecting data without utilizing a regression model. During the first simulation iteration of 200 days, we use the Daganzo-approximation (see Eq. 1) to estimate the routing costs and

support customer selection decisions. With the obtained data from the first iteration, we can train and subsequently use the regression model. In the next simulation iterations, we can use our trained approximation model for the customer selection decision. After every simulation iteration of 200 days, we add new data to our data set to further improve our approximation with more observations.

We simulate a finite horizon of 200 days and only plan for the upcoming day. We test four different spatial settings in both a backordering variant and a lost sales variant. Backordering in this case means that customer demand can be postponed indefinitely. For the lost sales case, a customer that arrived at the start of day t can only be postponed once, so to day $t + 1$. If not selected for delivery at the start of day t or day $t + 1$, the customer sales are lost. For all lost sales variants, we do not consider a vehicle capacity (Q), but a maximum vehicle distance (V) as a constraining factor, which mimics the situation of limited working hours. With distance as the constraining factor, the effect of customer selection on the number of lost sales can be better illustrated, since more efficient routing will allow for an increase in the number of customers that can be served. We include the vehicle distance constraint only for the lost sales case as it is more computationally demanding in our simulation to check this constraint.

After every simulation iteration, we retrain our model on the newly obtained data, including previously collected data, and test the model on a separate left-out validation data set. We include both old and new data in the training set to increase the number of observations and cancel out possible fluctuations in the data. Aside from regression model performance indicators, we store the traveled distances per day, the number of served customers, and lost sales ratios after a simulation run. We define the lost sales ratio as the percentage of lost sales compared with the total number of customers: number of lost sales/total number of customers.

Summarizing the solution structure, we start with a customer selection Phase 1, where a subset of customers is selected based on our distance approximation. More precisely, customers are sequentially selected based on the distance approximation. Customers are selected until the total fleet capacity has been reached, i.e., total fleet capacity = $K \cdot Q$, or in case of lost sales, we insert customers until the predicted distance exceeds the maximum fleet distance, i.e., maximum fleet distance = $K \cdot V$. Next, in Phase 2, a routing schedule is made for the selected customers. We apply a cluster-first-route-second heuristic by first selecting seeds based on: (i) the maximum distance from the depot and (ii) the maximum distance from the other seeds. Finally, a parallel assignment heuristic assigns customers to seeds and a TSP is solved for each vehicle. For solving the TSPs, we use a nearest neighbor heuristic for an initial solution, after which we run a 2-opt local search. Distances between locations are computed with the Euclidean-distance formula. After the routing schedule has been constructed, we move to the next day $t + 1$ and make a new customer selection decision, considering (i) the customers that were postponed the previous day t and (ii) the newly arrived customers for day $t + 1$.

For both the backordering case and the lost sales case, a feasibility check is needed after the customer selection phase, before a customer is definitely inserted into the route. Customers might have been assigned to a cluster seed, but cannot be inserted into the schedule due to the restraining vehicle capacity or maximum vehicle distance. In that case, we reject all left-over customers and they are again considered the next day.

We test our regression model on four different spatial instance types. The first three instance types are situated on a 100×100 grid. Instance type R randomly scatters customers on the grid. Instance type RC partially clusters customers by first randomly generating A cluster centrepoint locations, and next assigning customers with a probability $P(A)$ to a cluster

centrepoint, and assigning them a random location on the 100×100 grid with probability $1 - P(A)$. When assigned to a cluster centrepoint, customers are randomly placed within a radius r of the cluster centrepoint, with $r = 10$. We use two different settings for the RC instances: (i) generate 3 clusters ($A = 3$) and assign customers to a random cluster with probability $P(A) = 0.7$ or assign them to a random location with probability $1 - P(A) = 0.3$, and (ii) generate 8 clusters ($A = 8$) and assign customers to a random cluster with probability $P(A) = 0.7$ or assign them to a random location with probability $1 - P(A) = 0.3$. See Fig. 3 for an illustration of the R and RC instance types. For all three instance types, we consider a single depot located at $(50, 50)$.

The fourth instance type considers a special VRP-instance with multiple regions that are served by a single depot. Instead of the 100×100 grid, the total area is enlarged to a 200×200 grid. The depot is located at $(100, 100)$ and most customers (approximately 75%) are located inside the original area closest to the depot, Region 1 (see Fig. 4). Region 2 contains approximately 25% of the customers. The number of daily customer arrivals $|C_t|$ follows a Poisson distribution with $\lambda = 30$. This means that the number of customer arrivals fluctuates more heavily compared to the R and RC instances, on most days all customer demand can be fulfilled and customer selection is primarily needed on busy days with more arrivals.

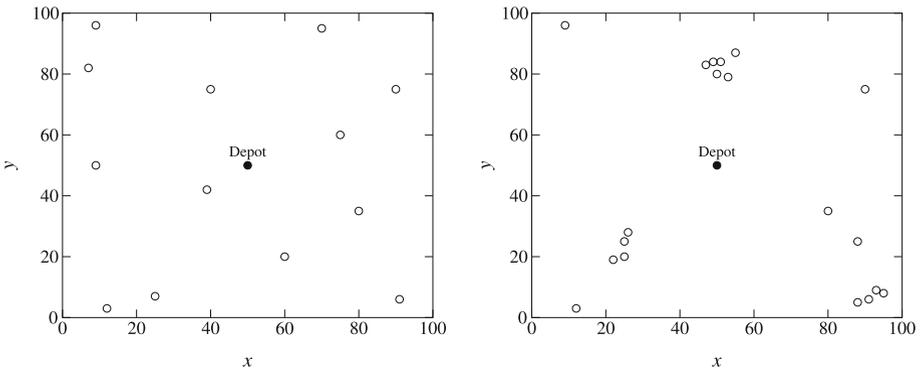
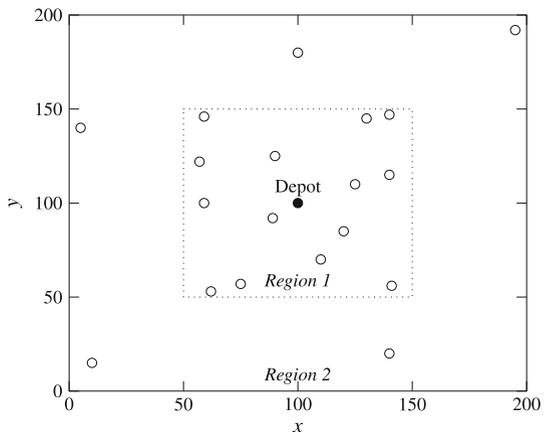


Fig. 3 Illustrative VRP instance with randomly scattered customers (left) and clustered customers ($A = 3$, $P(A) = 0.7$) (right)

Fig. 4 Illustrative VRP instance of a multi-region instance with backordering and vehicle capacity constraints



In the second case setting, we consider the four spatial instance types in a configuration with lost sales, i.e., the variant where customers that are not selected for delivery on their arrival day t or the next day $t + 1$, will leave and the sales are lost. This means that customer orders that arrive on day t , and are not served on day t , are automatically scheduled for the next day $t + 1$. If there are more customers postponed to day $t + 1$ than capacity is available, the surplus of postponed customers will be lost. Because of the variability in customer arrivals, on some days no decision is needed from our solution method, since the complete fleet capacity is consumed by previous-day orders. Thus, the decision method has less opportunity to make an impact on routing costs.

The multi-region instance type has been adapted slightly for the lost sales case. Since the furthest possible customer is located at the corner of the area, e.g., location (200,200), a vehicle needs to have a distance capacity V of at least 283 to make the trip from the depot to the customer and back. This extended distance capacity results in a situation where all customers close to the depot can easily be served, limiting the need for customer selection. Therefore, we reduced the number of vehicles K to 2.

All instance settings of the four instance types in both the normal and lost sales configuration are summarized in Table 8. The aim of the experiments with this stylized case is to examine if our proposed method can better recognise clustered groups of customers and group these together in a VRP schedule. Our features might be able to recognize an isolated customer in a sparsely populated area and postpone delivery until more customers arrive. Furthermore, we use different spatial instance types to examine the performance of our approximation models compared to the Daganzo-approximation benchmark. The lost sales case shows a different setting that causes a smaller decision space, which means that a single selection decision has more impact. Also, it shows the capabilities of our regression model to recognise and insert customers close to customers that are prioritized. A more efficient routing will potentially increase the number of served customers and decrease lost sales for the distance constrained lost sales experiments.

Table 8 Instance settings for backorder and lost sales instances

	R	RC	Multi-region
Number of Arriving customers	$U[40, 60]$	$U[40, 60]$	Poisson($\lambda = 30$)
Number of initial customers (C_0)	25	25	15
Customer demand $\sim U[a_d, b_d]$	$U[9, 11]$	$U[9, 11]$	$U[4, 6]$
Number of vehicles (K)	5	5	2
Vehicle capacity (Q)	80 (backordering) N/A (lost sales)	80 (backordering) N/A (lost sales)	80 (backordering) N/A (lost sales)
Maximum vehicle distance (V)	N/A (backordering) 150 (lost sales)	N/A (backordering) 150 (lost sales)	N/A (backordering) 283 (lost sales)
Number of clusters (A)	N/A	3 or 8	N/A
Probability of clustering ($P(A)$)	N/A	0.7	N/A

4.2 Settings for the waste collection case

We consider the dynamic collection of waste from underground containers in Amsterdam, The Netherlands, as depicted in Fig. 5. The Amsterdam waste collection problem can be considered as an Inventory Routing Problem (IRP), where we have to decide which containers to empty on which day, and how to route our vehicles to visit these containers. Here we specifically focus on the collection of household waste from 7995 containers in Amsterdam. For illustrative reasons, we focus in our experiments on the Southeast district of Amsterdam. This district is a secluded part of the city that consists of 353 underground heterogeneous containers, one depot, and two satellite locations. The containers are scattered over an area of 21.7 km². The daily waste disposal at each container c is stochastic and modeled using a Gamma-distribution, given by $d_c \sim \text{Gamma}(k_c, \theta_c)$, as common for these types of problems (Mes et al., 2014). We assume a homogeneous fleet of vehicles. Key performance indicators for comparing models are the service level and the distance traveled per ton of collected waste. The service level is dependent on the overflow of containers. An overflowed container has a fill level higher than the container capacity. We define the service level as the proportion of containers that are emptied on time, without overflow: $1 - (\text{number of overflowed containers} / \text{number of emptied containers})$.

We use a rolling horizon planning approach, where decisions are made on consecutive days t over a finite horizon $\mathcal{T} = \{1, \dots, T\}$. Each day, we plan for T days ahead, but only the decisions of $t = 1$ are fixed. To be able to solve problem sizes of up to 7995 containers in reasonable time, we propose a solution methodology consisting of the following three phases: (i) container selection, (ii) day assignment, and (iii) route construction. The first phase concerns the selection of containers based on overflow probabilities of every container. When the

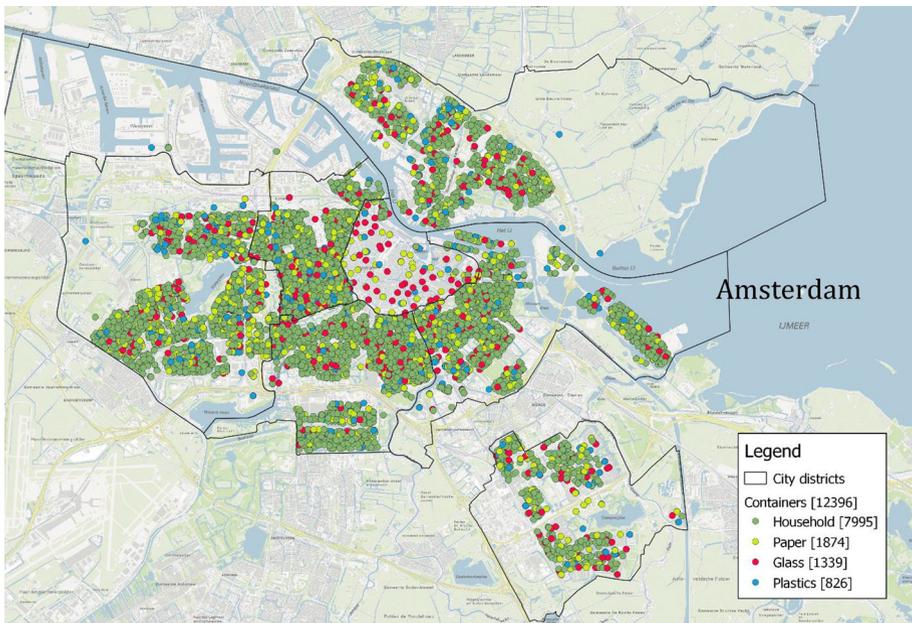


Fig. 5 Map of the underground waste containers in Amsterdam, The Netherlands (source: maps.amsterdam.nl)

overflow probability exceeds a certain threshold, the respective container is considered for the next phase. The second phase concerns the planning of collection days for the pre-selected containers. In this phase, both the service level and the travel costs are considered, i.e., both the time and space dimension of the inventory routing problem. The time dimension of an IRP concerns the timing of container emptying and the amount of waste collected from each respective container, the space dimension concerns the routing along the selected containers. The third phase concerns the construction of routes for the first day ($t = 1$) of the planning horizon. We use a cluster-first-route-second approach, which constructs routes in four steps: (i) clustering containers using adapted k-means, (ii) feasible sequencing using nearest insertion, (iii) combining sequences into feasible routes, and (iv) improving the feasible solution using a 2-opt metaheuristic. See Heijnen (2019) for more details on the route construction phase.

Our proposed method concerns the approximation used in the second phase: the allocation of containers to days. We use an algorithm that iteratively assigns containers to days based on the distance approximation. In the next section, we explain how our regression model can be adapted to predict a cost function with multiple objectives, i.e., distance and service level. We benchmark our method using the Daganzo-approximation for the distance approximation (see Eq. 1). Our regression model predicts a combined cost term, including distance and service level. However, the Daganzo-approximation only considers distance. Therefore, we adapt the benchmark method to consider a combined cost term including the Daganzo-approximation and a penalty factor for emptying a container c too early or too late:

$$\text{Selection costs}_c = \text{Daganzo}(c) + \text{Timing penalty}(c). \quad (11)$$

The timing penalty is determined using the following:

$$\text{Timing penalty}(c) = \begin{cases} \text{Too early penalty,} & \text{if } t < EOD - 1, \\ 0, & \text{if } t = EOD - 1, \\ \text{Too late penalty,} & \text{if } t > EOD - 1, \end{cases} \quad (12)$$

with t being the day that is considered for the assignment and EOD being the expected overflow day.

The expected overflow day is determined using an overflow probability, determined using the expected fill levels that are estimated with the probability density function of the Gamma-distribution. Since container overflow needs to be prevented, we set an acceptable overflow probability (AOP). So, the EOD is the day before the probability of overflow exceeds the AOP. In our experiments, we use different levels for the AOP.

4.3 Adaptations to the generic model for the waste collection case

The waste collection problem and other IRPs differ from the standard VRP by being multi-objective: the distance needs to be minimized and the service level should be maximized (or attain a certain threshold). Practical problems will arise when there is too much overflow of containers. For our implemented benchmark method (the Daganzo-approximation), we separately assess the service level requirements by adding a timing penalty (see Eq. 12) to the approximated distance. For our new approximation model, we can combine the performance indicators by both approximating the distance and the service level together.

We introduce two new features to estimate the actual service level, namely the service level calculated using the expected fill levels (F49) and the average expected fill level of containers as a percentage of the container capacity (F50). For both features, we use the known container

capacities to calculate the feature values. F49 can be calculated by considering for each container the days until last emptying, the average waste disposal per day for this container, and its capacity. In preliminary experiments, we observe that F49 can estimate the service level reasonably well; when tested in a single-feature regression model aimed at predicting the service level, we observe a relative mean absolute error of 9.3%. F50 is an error term that is added to take into account possible deviations from the expected fill levels: when the demand of a container is close to its capacity, it has a high chance of overflow. Thus, in case of equal expected service level and distance, the container with a higher fill level is favored for emptying on the current day.

After scaling both target variables to the domain $[0, 1]$, we define a new cost function (13), that combines distance and service level terms in one objective function. The regression model estimates the costs, i.e., it is trained to predict the value of $\zeta_{c,t}$.

$$\zeta_{c,t}(S_t, x_{c,t}) = w^d \cdot d_{S_t, x_{c,t}} + w^\alpha \cdot \alpha_{S_t, x_{c,t}}, \quad \forall c \in C : C \subseteq I, \forall t \in \mathcal{T}, \quad (13)$$

with $\zeta_{c,t}$ being the combined cost for inserting container $c \in C$ on day $t \in \mathcal{T}$. C is the set of containers that are not yet inserted, I is the complete set of containers that has been pre-selected in phase 1 of the algorithm, so $C \subseteq I$. S_t is the current state from which we derive the feature values for the already selected set of containers for day t , and $x_{c,t}$ is the decision to insert container c on day t . The costs are determined using the predicted distance d and service level α . The weights w strike a balance between the importance of the distance and the service level. In our experiments, we adjust both weights.

5 Computational experiments and results

In this section, we discuss our experiments and results for the stylized customer selection case and the waste collection case. This section (i) illustrates the use of our proposed distance approximation model in a decision-support context, and (ii) shows the use of our adaptive learning framework, as discussed in Sect. 3.7. We start with discussing the results for the stylized customer selection case in Sect. 5.1, first for the backordering configuration, then for the lost sales case. In Sect. 5.2, we discuss the results for the dynamic waste collection case of Amsterdam. To ease the presentation, we only show the results for the linear regression model for both cases, as its performance is relatively close to those of the more advanced models, i.e., random forests, lightGBM, or neural networks, see Sect. 3.6.

5.1 Results for the stylized customer selection case

We first validate our regression model on the stylized customer selection case and show the application of the adaptive learning feedback loop. For this, we create a simulation model in Python. In our experiments, we conduct 15 iterations (N) of the adaptive learning feedback loop, i.e., in one iteration we (i) use a predictive model to make customer selection decisions, (ii) construct routing schedules, (iii) retrain the predictive model on the data, and repeat the process in the next iteration. After each iteration of the adaptive learning feedback loop, we forget the old data, since we obtain enough training data in a single iteration. To report statistically significant results, we conduct several replications over these 15 iterations. We determine the number of replications by calculating the relative error of the total distance over the replications for each instance type. To obtain a relative error less than 5% with 95% confidence, 10 replications are needed for all instance types.

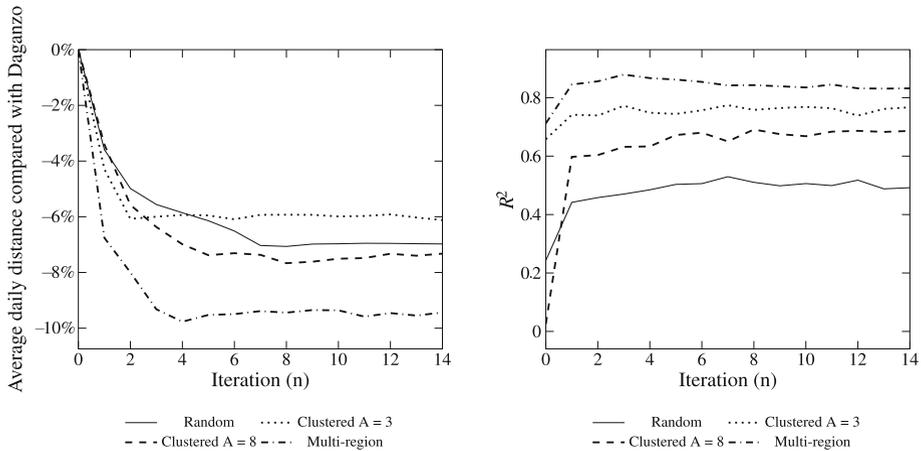


Fig. 6 Experimental results of the randomly scattered (R), clustered (RC), and multi-region instances under backordering: average daily distance compared with the Daganzo-approximation (left) and the R^2 on a left-out validation set (right), using 15 iterations and 10 replications

Two policies are compared: (i) the benchmark Daganzo-approximation and (ii) our proposed regression model using 23 features. For our proposed model, in iteration 0, the daily decision to serve or postpone a customer is made on the basis of the Daganzo-approximation. In iterations 1 to 14, we use our proposed regression model. Every iteration consists of 200 execution days. The reported statistics are the averages from 10 replications and are relative to the performance of the Daganzo-approximation benchmark. We report significance by performing paired t-tests.

Figure 6 shows the results for the backordering case. The left graph shows the average daily VRP-distance over the 15 iterations, compared to the Daganzo-approximation (0% line).

The difference between the Daganzo-approximation and the approximation models, in terms of distance, is significant for all instances with 95% confidence. A negative percentage indicates a saving of the regression model in comparison with the Daganzo-approximation. The number of served customers per day using the Daganzo-approximation or the regression model are similar. The right figure indicates the prediction quality of the regression model, expressed with the R^2 , reported after every iteration on a separate validation data set. Both figures indicate that the performance on the random instances is the lowest, subsequently followed by the clustered instances and the multi-region instances. We observe that when customers are more scattered, the performance gains are lower because the variance between customers is large and cannot be fully captured by our features. Nevertheless, for all instances we improve on the Daganzo-approximation with a reduction of 6.1% up to 9.4% in average daily distance after 15 iterations.

Next, we discuss the results for the lost sales case. Figure 7 shows the average daily distance per customer compared to the Daganzo-approximation (0% line) (left), and the R^2 of the regression model per simulation iteration (right). The difference between the Daganzo-approximation and the approximation models, in terms of distance, is significant

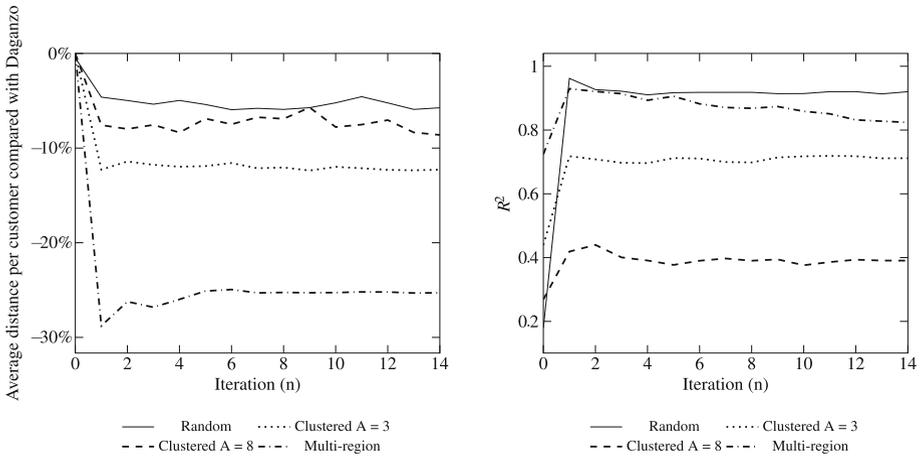


Fig. 7 Experiment results of the randomly scattered (R), clustered (RC) and multi-region instances with lost sales: average distance per customer compared with the Daganzo-approximation (left) and the R^2 on a left-out validation set (right), using 15 iterations and 10 replications

for all instances with 95% confidence. Especially the multi-region case shows a considerable improvement (25.3% after 15 iterations), which can be explained by a better adaptation to the lost sales configuration, compared to the static Daganzo-formula. In addition, we observe that the fluctuating Poisson arrivals change the balance in the system: on most days, a large percentage of all customers can be served, but on days with peak demand, the decision on what customers to serve becomes more important and our approximation model, opposed to the Daganzo-approximation, chooses to prioritize close-by customers and lose far away customers. For the other instance types (random and clustered), we observe similar or slightly larger savings compared to the backordering configuration, the savings in average distance per customer range from 5.7% up to 12.3% after 15 iterations. The R^2 on the clustered instances is slightly lower than before, but this is not directly reflected in the performance of the models.

Figure 8 shows more performance statistics for the lost sales case. The average number of served customers (left) and lost sales ratio (right) are compared with the Daganzo-approximation (0% line). We observe that when using our approximation model for customer selection, we can serve more customers, except for the random and multi-region instance type, which show a lower and similar number of served customers compared with the Daganzo-approximation, respectively. The lost sales ratios are, for both the regression model and the Daganzo-approximation, negligible. Nevertheless, the difference between the Daganzo-approximation and our model is significant at a confidence level of 95%, with as exception the difference on the clustered (A=3) instances. We observe that the Daganzo-approximation almost always has a better lost sales ratio. Our approximation model often chooses to neglect expensive lost sales customers, where the Daganzo-approximation does select these customers. As a result, the Daganzo-approximation has less lost sales, but needs to travel longer distances compared with our approximation model.

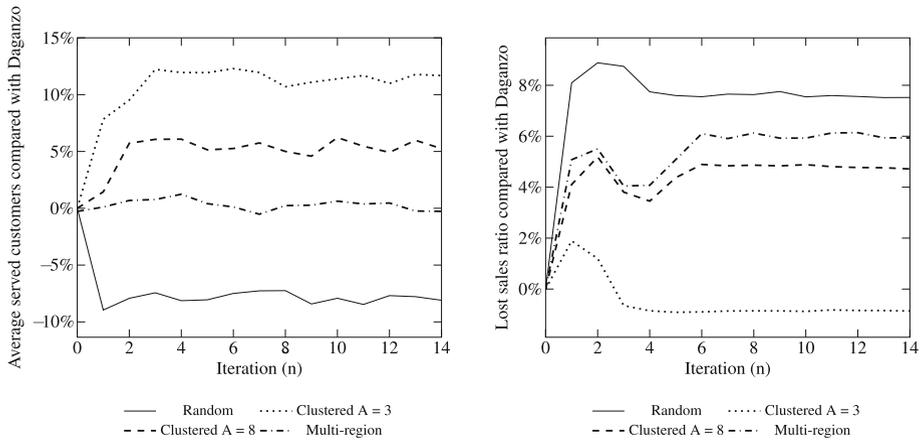


Fig. 8 Experiment results of the randomly scattered (R), clustered (RC) and multi-region instances with lost sales: average number of served customers compared with the Daganzo-approximation (left) and the number of lost sales compared with the Daganzo-approximation (right), using 15 iterations and 10 replications

5.2 Results for the waste collection case

We created a discrete-event simulation model in Java, with two types of actors: the inhabitants who dispose waste in containers and the waste collectors who empty the containers. For simplicity, we only focus on the planning phase and ignore possible disruptions during the execution of routes. At the beginning of each day, the three-phase planning procedure is executed to plan the waste collection routes for the corresponding day (see Sect. 4.2). We use a rolling horizon of three days, which is found to strike the best balance between performance and computational efficiency. For our simulation, we use a simulation run length of 125 days with a 25-day warmup period. Given the relatively long run length, it appears that 3 replications are enough to obtain a relative error of at most 5% using a 95% confidence interval for the total driving distance.

Three policies are compared: (i) the benchmark Daganzo-approximation with a service level penalty, (ii) our proposed regression model, which combines distance and service level approximations, and (iii) a myopic policy that uses a horizon of $T = 1$ and always favors the containers with the highest expected fill levels. For both the benchmark policy and our regression model, the respective overflow penalty and approximation weights can be tuned. The tuning of these parameters can shift the focus, either favoring the service level or distance. Table 9 summarizes the relevant experimental parameters for each model. The acceptable overflow penalty (AOP) is only used for the Daganzo-approximation since the myopic

Table 9 Experimental parameters

Policy	Planning horizon	AOP	(w^d, w^α)
Myopic	1	–	–
Daganzo	3	{0.1,0.2,0.3}	–
Regression	3	–	{(1,1),(1,10),(10,1)}

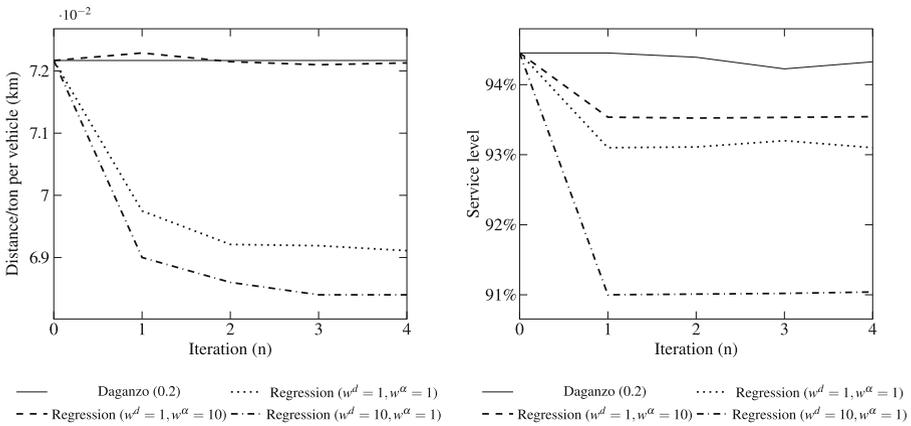


Fig. 9 Performance of approximation policies over several iterations, AOP = 0.2 (Daganzo) and $(w^d, w^\alpha) = \{(1, 1), (1, 10), (10, 1)\}$ (Regression), $N = 4, 3$ replications, 125-day horizon with 25-day warmup period

method only considers container fill levels and our regression model has its own service level approximation.

The implemented model for the case contains 18 features and is trained using the VRP data obtained from the waste collection case. Although the demand for waste collection is stochastic, the system is stable, i.e., there are no external disruptions and the parameters for the demand, modeled with the Gamma-distribution, do not change. Nevertheless, it might still be the case that the approximation can be improved using our proposed adaptive learning framework, i.e., (i) using a predictive model to make container selection decisions, (ii) construct routing schedules, and (iii) retrain the predictive model and repeat the process. After the first iteration of this process, we forget the old data, since we obtain enough training data in one iteration. To obtain the initial training data for our regression model, we first use the Daganzo-approximation (see Eq. 1) for the customer selection in combination with a service level penalty factor, as described in Sect. 4.2. After enough training data (VRP realizations) has been obtained, we can train our regression model on the combined distance and service level term as presented in Sect. 4.3. Figure 9 shows the respective distance and service level for the three settings of the regression model during several iterations of the adaptive learning feedback loop. The performance of the best setting for the Daganzo-approximation is also shown. We perform a paired t-test for the two performance indicators as observed in iteration 4, comparing the regression models with the Daganzo-approximation. The paired t-test confirms that the difference in distance per vehicle and service level is significant for all models with 95% confidence, with as exception the difference between the Daganzo-approximation and Regression with parameters $w^d = 1$ and $w^\alpha = 10$.

We observe that the weights in the cost function $\zeta_{c,t}$ have an effect on the performance of the model. When the weight for the distance (w^d) is relatively low, the model favors high service levels over distance reduction, and vice versa. The improvement over the iterations is limited, which indicates that for this case, the initial training on the training set as described at the beginning of this section, was sufficient.

A more detailed comparison of all experiments can be found in Table 10. First, the added value of a rolling planning horizon is confirmed by the poor performance of the myopic policy, in comparison with the Daganzo-approximation and the regression model: the service level is relatively low, and the distance is over 15% more in comparison with the worst

Table 10 Performance of approximation policies for all experiments

Policy	km/ton per vehicle	Service level	Nr. of vehicles
Myopic	0.0959	86.4%	5
Daganzo (0.1)	0.0830	96.5%	4
Daganzo (0.2)	0.0722	94.1%	4
Daganzo (0.3)	0.0827	90.5%	4
Regression ($w^d = 1, w^\alpha = 1$)	0.0691	93.2%	4
Regression ($w^d = 1, w^\alpha = 10$)	0.0721	93.5%	4
Regression ($w^d = 10, w^\alpha = 1$)	0.0684	91%	4

performing approximation method. Compared with the best performing approximation method, the myopic policy is more than 28% worse. Also, an additional vehicle is needed. Furthermore, we observe that the regression model results in a better performance compared to using the Daganzo-approximation: there is an improvement in distance ranging from 0.13% to almost 17%, compared with similar or slightly worse service levels. The difference, in terms of distance and service level, between the best performing Daganzo-method and the best performing regression model is significant with 95% confidence.

6 Conclusions

We developed a distance approximation method to support customer selection, encompassing a large range of temporal and spatial features. This method can be used to predict distance and service levels within transportation problems, for use in customer assignment and selection problems, i.e., the assignment of customers to days for a multi-period vehicle routing problem or supporting fast customer selection decisions in situations with limited capacity. We illustrated the approach considering two relatively large vehicle routing cases: a fictional case with multiple spatial settings for both backordering and lost sales, and a real case of dynamic waste collection in Amsterdam, The Netherlands. As a benchmark, we implemented the Daganzo-approximation.

The new distance and service level approximation model was introduced in such a way that it can be applied to a wide range of problems. We showed which features have the highest importance for TSP and VRP models, showed the performance gain of our model compared with well-known closed-form distance approximation formulas, showed that we can predict distance fairly accurately without solving the TSP or VRP, explained the automatic feature selection methods for linear regression and tree-based methods, and illustrated the use of an automatic hyperparameter tuning approach for the tree-based methods and neural networks. We described the approach of combining offline learning with online optimization, and how to iteratively update or improve the approximations. Finally, we validated our machine learning model on the stylized customer selection case and the multi-period waste collection case with stochastic demands. The stylized case showed that the approximation models can be successfully utilized for customer selection problems with different spatial settings. We showed how our customer selection method, utilizing the distance approximation model, was applied to a real case of the waste collection planning of Amsterdam, The Netherlands. The application of our model to an inventory routing problem (IRP) showed a different practical

setting for which our proposed method can be used. For both the stylized case and the waste collection case, our proposed model performs reasonably better than the benchmark policy. Also, we showed that our regression model shows better performance on clustered instances and complex structure instances than on randomly scattered instances.

Further research can be done on new features that describe certain vehicle routing problem instances more specifically. Especially the rectangular partitioning structure provides opportunities for the design of new features, e.g., the rectangles can be given weights corresponding to the number of customers inside them, and different ways of partitioning could be explored, e.g., using adaptive grids that automatically identify customer clusters. We would like to stress that computational effort is an important factor in calculating features, especially when the approximation needs to be done often and relies on its speed compared with solving a TSP or VRP using heuristics. A limitation of our distance approximation model is its inability to look ahead, since it aims to minimize the costs of the current day only. For the waste collection case, we used a rolling horizon approach, but we believe more research could be done on the inclusion of a look-ahead policy into our distance approximation, e.g., by utilizing (deep) reinforcement learning methods that can minimize long term costs using features as those proposed in this paper.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Agatz, N., Campbell, A., Fleischmann, M., & Savelsbergh, M. (2011). Time slot management in attended home delivery. *Transportation Science*, *45*(3), 435–449.
- Akkerman, F., Mes, M., & Heijnen, W. (2020). Distance approximation for dynamic waste collection planning. In E. Lalla-Ruiz, M. Mes, & S. Voß (Eds.), *Computational Logistics* (pp. 356–370). Cham: Springer.
- Aras, N., Aksent, D., & Tuğrul Tekin, M. (2011). Selective multi-depot vehicle routing problem with pricing. *Transportation Research Part C: Emerging Technologies*, *19*(5), 866–884.
- Archetti, C., Fernández, E., & Huerta-Muñoz, D. L. (2017). The flexible periodic vehicle routing problem. *Computers & Operations Research*, *85*, 58–70.
- Arnold, F., & Sörensen, K. (2019). What makes a vrp solution good? the generation of problem-specific knowledge for heuristics. *Computers & Operations Research*, *106*, 280–288.
- Baita, F., Ukovich, W., Pesenti, R., & Favaretto, D. (1998). Dynamic routing-and-inventory problems: A review. *Transportation Research Part A: Policy and Practice*, *32*(8), 585–598.
- Bard, J. F., & Nananukul, N. (2009). Heuristics for a multiperiod inventory routing problem with production decisions. *Computers & Industrial Engineering*, *57*(3), 713–723. <https://doi.org/10.1016/j.cie.2009.01.020>
- Bard, J. F., Huang, L., Jaillet, P., & Dror, M. (1998). A decomposition approach to the inventory routing problem with satellite facilities. *Transportation Science*, *32*(2), 189–203.
- Beardwood, J., Halton, J. H., & Hammersley, J. M. (1959). The shortest path through many points. *Mathematical Proceedings of the Cambridge Philosophical Society*, *55*(4), 299–327. <https://doi.org/10.1017/S0305004100034095>
- Beliën, J., De Boeck, L., & Van Ackere, J. (2014). Municipal solid waste collection and management problems: A literature review. *Transportation Science*, *48*(1), 78–102.

- Benjamin, A., & Beasley, J. (2010). Metaheuristics for the waste collection vehicle routing problem with time windows, driver rest period and multiple disposal facilities. *Computers & Operations Research*, 37(12), 2270–2280.
- Braekers, K., Ramaekers, K., & Van Nieuwenhuysse, I. (2016). The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*, 99, 300–313.
- Brochu, E., Cora, V. M., & de Freitas, N. (2010). A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *CoRR arXiv:1012.2599* [abs].
- Caceres-Cruz, J., Arias, P., Guimaran, D., Riera, D., & Juan, A. A. (2014). Rich vehicle routing problem: Survey. *ACM Computing Surveys*, 47(2), 1–28.
- Chien, T. (1992). Operational estimators for the length of a traveling salesman tour. *Computers & Operations Research*, 19(6), 469–478. [https://doi.org/10.1016/0305-0548\(92\)90002-M](https://doi.org/10.1016/0305-0548(92)90002-M)
- Christofides, N., & Eilon, S. (1969). Expected distances in distribution problems. *Journal of the Operational Research Society*, 20, 437–443.
- Coelho, L. C., Cordeau, J. F., & Laporte, G. (2014). Thirty years of inventory routing. *Transportation Science*, 48(1), 1–19.
- Dror, M., Laporte, G., & Trudeau, P. (1994). Vehicle routing with split deliveries. *Discrete Applied Mathematics*, 50(3), 239–254.
- Duhamel, C., Santos, A. C., & Aloise, D. J. (2009). Multicommodity formulations for the prize collecting vehicle routing problem in the petrol industry. Research report, Université Blaise Pascal (Clermont Ferrand 2).
- El-Hajj, R., Guibadj, R. N., Moukrim, A., & Serairi, M. (2020). A PSO based algorithm with an efficient optimal split procedure for the multiperiod vehicle routing problem with profit. *Annals of Operations Research*, 291(1), 281–316.
- Figliozzi, M. A. (2008). Planning approximations to the average length of vehicle routing problems with varying customer demands and routing constraints. *Transportation Research Record*, 2089(1), 1–8. <https://doi.org/10.3141/2089-01>
- Fisher, M. L., & Jaikumar, R. (1981). A generalized assignment heuristic for vehicle routing. *Networks*, 11(2), 109–124. <https://doi.org/10.1002/net.3230110205>
- Gromicho, J., van Hoorn, J., Kok, A., & Schutten, J. (2012). Restricted dynamic programming: A flexible framework for solving realistic vrps. *Computers & Operations Research*, 39(5), 902–909.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference and prediction* (2nd ed.). Berlin: Springer.
- Head, T., Kumar, M., Nahrstaedt, H., Louppe, G., & Shcherbatyi, I. (2021). Scikit-optimize/scikit-optimize. <https://doi.org/10.5281/zenodo.5565057>.
- Heijnen, W. (2019). Improving the waste collection planning of amsterdam. <http://essay.utwente.nl/78290/>.
- Hindle, A., & Worthington, D. (2004). Models to estimate average route lengths in different geographical environments. *Journal of the Operational Research Society*, 55(6), 662–666. <https://doi.org/10.1057/palgrave.jors.2601751>
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 3146–3154.
- Keany, E. (2020). *BorutaShap: A wrapper feature selection method which combines the Boruta feature selection algorithm with Shapley values*. <https://doi.org/10.5281/zenodo.4247618>
- Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *CoRR arXiv:1412.6980* [abs]
- Kursa, M. B., & Rudnicki, W. R. (2010). Feature selection with the Boruta package. *Journal of Statistical Software*, 36(11), 1–13. <https://doi.org/10.18637/jss.v036.i11>
- Kwon, O., Golden, B., & Wasil, E. (1995). Estimating the length of the optimal tsp tour: An empirical study using regression and neural networks. *Computers & Operations Research*, 22(10), 1039–1046. [https://doi.org/10.1016/0305-0548\(94\)00093-N](https://doi.org/10.1016/0305-0548(94)00093-N)
- Lalla-Ruiz, E., & Voß, S. (2020). A POPMUSIC approach for the multi-depot cumulative capacitated vehicle routing problem. *Optimization Letters*, 14(3), 671–691.
- Mes, M. (2012). Using simulation to assess the opportunities of dynamic waste collection. In S. Bangsow (Ed.), *Use cases of discrete event simulation: Appliance and research* (pp. 277–307). Berlin Heidelberg, Berlin: Springer.
- Mes, M., Schutten, M., & Rivera, A. P. (2014). Inventory routing for dynamic waste collection. *Waste Management*, 34(9), 1564–1576.
- Moin, N. H., & Salhi, S. (2007). Inventory routing problems: A logistical overview. *Journal of the Operational Research Society*, 58(9), 1185–1194.

- Mourgaya, M., & Vanderbeck, F. (2007). Column generation based heuristic for tactical planning in multi-period vehicle routing. *European Journal of Operational Research*, 183(3), 1028–1041. <https://doi.org/10.1016/j.ejor.2006.02.030>
- Nicola, D., Vetschera, R., & Dragomir, A. (2019). Total distance approximations for routing solutions. *Computers & Operations Research*, 102, 67–74.
- Novoa, C., & Storer, R. (2009). An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. *European Journal of Operational Research*, 196(2), 509–515.
- Orlenko, A., & Moore, J. H. (2021). A comparison of methods for interpreting random forest models of genetic association in the presence of non-additive interactions. *BioData Mining*, 14(1), 9. <https://doi.org/10.1186/s13040-021-00243-0>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Powell, W. B., & Ryzhov, I. O. (2013). *Optimal learning and approximate dynamic programming* (pp. 410–431). New York: Wiley.
- Rasku, J., Kärkkäinen, T., & Musliu, N. (2016). Feature Extractors for Describing Vehicle Routing Problem Instances. In B. Hardy, A. Qazi, S. Ravizza (Eds.), *5th student conference on operational research (SCOR 2016), Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, OpenAccess Series in Informatics (OASICs)* (Vol. 50, pp. 7:1–7:13).
- Robusté, F., Daganzo, C. F., & Souleyrette, R. R. (1990). Implementing vehicle routing models. *Transportation Research Part B: Methodological*, 24(4), 263–286.
- Robusté, F., Estrada, M., & López-Pita, A. (2004). Formulas for estimating average distance traveled in vehicle routing problems in elliptic zones. *Transportation Research Record*, 1873(1), 64–69. <https://doi.org/10.3141/1873-08>
- Roldán, R. F., Basagoiti, R., & Coelho, L. C. (2016). Robustness of inventory replenishment and customer selection policies for the dynamic and stochastic inventory-routing problem. *Computers & Operations Research*, 74, 14–20. <https://doi.org/10.1016/j.cor.2016.04.004>
- Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operation Research*, 35, 254–265.
- Taslimi, M., Batta, R., & Kwon, C. (2020). Medical waste collection considering transportation and storage risk. *Computers & Operations Research*, 120, 104966. <https://doi.org/10.1016/j.cor.2020.104966>
- Tricoire, F., Romauch, M., Doerner, K. F., & Hartl, R. F. (2010). Heuristics for the multi-period orienteering problem with multiple time windows. *Computers & Operations Research*, 37(2), 351–367. <https://doi.org/10.1016/j.cor.2009.05.012>
- Ulmer, M. W., Mattfeld, D. C., & Köster, F. (2018). Budgeting time for dynamic vehicle routing with stochastic customer requests. *Transportation Science*, 52(1), 20–37.
- Ulmer, M. W., Goodson, J. C., Mattfeld, D. C., & Hennig, M. (2019). Offline-online approximate dynamic programming for dynamic vehicle routing with stochastic requests. *Transportation Science*, 53(1), 185–202.
- Vidal, T., Maculan, N., Ochi, L. S., & Vaz Penna, P. H. (2016). Large neighborhoods with implicit customer selection for vehicle routing problems with profits. *Transportation Science*, 50(2), 720–734. <https://doi.org/10.1287/trsc.2015.0584>
- Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2), 301–320.