



# Distance Approximation for Dynamic Waste Collection Planning

Fabian Akkerman<sup>(✉)</sup>, Martijn Mes<sup>(✉)</sup>, and Wouter Heijnen

University of Twente, Department of Industrial Engineering and Business  
Information Systems, Enschede, The Netherlands  
f.r.akkerman@student.utwente.nl, m.r.k.mes@utwente.nl

**Abstract.** Approximating the solution value of transportation problems has become more relevant in recent years, as these approximations can help to decrease the computational effort required for solving those routing problems. 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 accurately using simple regression models and only a limited number of features. We use features found in scientific literature and introduce a new class of spatial features. The model is validated on a dynamic waste collection case in the city of Amsterdam, the Netherlands. We introduce a cost function that combines the travel distance and service level, and show that our model can reduce distances up to 17%, while maintaining the same service level, compared to a well-known heuristic approximation. Furthermore, we show the benefits of using approximations for combining offline learning with online or frequent optimization.

**Keywords:** Distance approximation · Vehicle routing · Inventory routing problem · Waste collection

## 1 Introduction

There is an increasing need among logistics companies for faster solution methods to solve vehicle routing problems. In general, faster solving times are needed to cope with the addition of new problem attributes, fast-changing demands, disruptions, and increasing problem sizes. There are numerous applications for which these approximations can be helpful. What time-slots for delivery should we offer our customer, considering the current set of customers and the location of the new customer? How should customers be re-clustered when new customers are added, removed, or vehicles break down? Is it profitable to accept a far-away customer to include in the current vehicle routes, or should we reserve time for future customers? For these, and other situations, an approximation can be used to obtain an indication of the costs, without solving the actual problem.

In this paper, we develop an approximation method that utilizes regression models to approximate the costs within transportation problems related to distance and service level. 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 [10]. We validate our model by applying it to a case study on dynamic waste collection in Amsterdam, the Netherlands. Fast approximation techniques can be advantageous in this case, as the problem size is huge, we have to consider a longer-term planning horizon, and demand is stochastic. The problem can be considered as an inventory routing problem, where we have to decide which containers to empty on which day, and how to route our vehicles to visit these containers. To reduce computational complexity, we split these decisions: first we select the waste containers to be emptied on the different days between the current day and the end of the horizon, and next decide on the routing of our vehicles for the current day. Even though we select the containers without solving vehicles routes, we use an approximation that predicts the costs and service levels for a selection of containers based on geographical information and fill levels of containers. To avoid excessive computation times in online or frequent decision making, we propose a combination of offline learning (training the approximation model) and operational decision making (VRP).

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, on waste collection problems, and describe our contribution. In Sect. 3, we describe the problem, introduce our approximation model and further discuss the combination of offline learning and online optimization. In Sect. 4, we validate and illustrate our model using the waste collection case. We close with conclusions and future research directions in Sect. 5.

## 2 Literature

In this section, we briefly review the literature on fast solution methods for the VRP. We first discuss the use of offline methods that support online or operational decision making, e.g., assigning customers to clusters and time slots before making a routing decision. Next, we treat the relevant literature about modelling the planning of waste collection related to our case study.

VRP research increasingly considers real-life, dynamic environments, which typically involve stochastic demands, stochastic travel times, and other disturbances [7]. This means that VRP models need to come up with robust plans that can handle changing environments. As opposed to exact solutions, approximation methods are generally better able to solve large problems and are typically more robust, hence they are often applied in real-life situations [8]. With increasing problem sizes or longer planning horizons, 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 [12]. 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 [14]. 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 [20].

Several authors use offline methods to improve online decisions. In [24], approximate dynamic programming (ADP) 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 [25]. 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 [19] 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 computational effort to an offline stage.

Aside from ADP, research has been conducted on the use of machine learning methods to approximate the value of a VRP decision. In [2], 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. The research in [18] is focused on the prediction of travel distance using linear regression, based on 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 [23] with clustered customers. For a comprehensive literature review on distance-approximation, we refer to [18].

The planning of waste collection is gaining a lot of attention in the scientific literature [5]. 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) [5]. WCVRPs have as objective to find an optimal route 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 [6]. 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 [3]. 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 prob-

lems like in [1], and (iii) model the planning of waste collection as an inventory routing problem (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) [9]. The IRP is a medium-term problem, in contrast to the short-term character of the regular VRP or WCVRP [1]. The classification in [13] 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 [4]. Much waste collection research is done both for single-period models and multi-period models. However, since the long-term planning approach has positive effects on long-term outcomes [17], contemporary research tends to treat multi-period models [13]. An application of the IRP to a waste collection problem can be found in [15].

Our contribution to the scientific literature is threefold. First, we develop a solution methodology that reduces the problem size of multi-period, dynamic VRPs. Second, we introduce several new spatial features to better estimate the characteristics of VRP solutions. Third, we illustrate our solution methodology using a dynamic waste collection case.

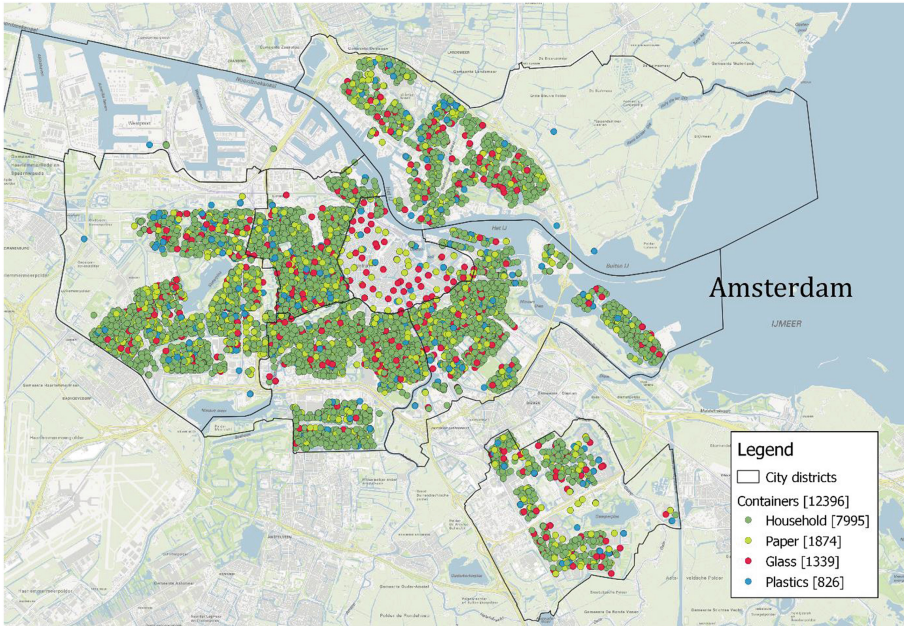
### 3 Problem Description and Formulation

In this section, we subsequently introduce our case study and the approximation model. With respect to the model, we make a clear separation between case-specific elements and generic 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. After the model description, we select features and evaluate the performance of both models on a test set. Next, we describe the adaptations of the generic model to apply it in our case study. We end this section with our framework for improving the approximation by combining online optimization and offline learning.

#### 3.1 A Solution Structure for the Waste Collection Case

We consider the dynamic collection of waste from underground containers in Amsterdam, the Netherlands, as depicted in Fig. 1. 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 km<sup>2</sup>. 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 type of problems [16]. 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:  $1 - (\text{number of overflowed containers}/\text{number of emptied containers})$ .



**Fig. 1.** Map of the waste collection containers in Amsterdam, The Netherlands (source: [maps.amsterdam.nl](http://maps.amsterdam.nl))

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 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. 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 [13] for more details on the route construction phase.

Our proposed method concerns the approximation to be used in the second phase: the allocation of containers to days. To benchmark our method, we consider a policy that sequentially builds a day assignment solution using the expected fill levels of containers for the time dimension, and the Daganzo-approximation for the space dimension, i.e., while iterating over all containers and days, it looks for the cheapest allocation of a container to a day, considering a combined term for service level and distance. The Daganzo-approximation is a fairly accurate approximation of a VRP distance [22] and is calculated as follows:

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

where  $k$  is an area shape constant,  $N$  the number of customers,  $C$  the maximum number of customers a vehicle can serve, and  $A$  is the area size. The time dimension is covered using a penalty factor based on an acceptable overflow probability (AOP). The AOP is a tunable threshold, which allows for giving penalties when containers are emptied later than their desired emptying date (DED).

### 3.2 A Generic Model for Approximating Distance

We now introduce our approximation model, starting with a basic TSP model and later extending it to a VRP model. Various features are proposed and evaluated using (i) linear regression, (ii) random forests, and (iii) neural networks. We generate the training and testing data using our simulation of the waste collection case (see Sect. 4). Each day in the simulation, we use the three-phase planning approach to select the containers to empty and subsequently solve the VRP, treating the container as customers. Hence, this choice of generating the vehicle routes as training data does not affect the generic applicability. For the purpose of creating training data for the TSP, we randomly select the route of a single vehicle for each day. For both the TSP and VRP routes, we store the locations, the fill levels, as well as the location of the depot and possible planned visits of satellite locations. Note that for the purpose of generating the training and testing instances, we only focus on planned routes, and ignore possible disruptions during the day, which will be considered later on in Sect. 4.

**Model for the Traveling Salesman Problem.** For the approximation of a TSP route, we only consider spatial features. The features are summarized in Table 1 and, if deemed relevant, further explained below.

F5–F7 are based on the smallest rectangle that can be fitted around the locations. F8–F12 are distance related features for which we compute the average distances to the depot, cluster centroid, and cluster midpoint. The angle related features F13–F15 express the dispersion of the customers by taking the variance of several different bearings between customer and the depot, cluster centroid, and cluster midpoint. The bearing  $\beta_{a,b}$  of two points  $(a, b)$  expressed in latitude and longitude, is the angle between the line connecting the two points and the north-south line of the earth, and can be calculated with (2), (3) and (4).

$$x = \cos(b_{lat}) * \sin(\Delta(a_{long}, b_{long})) \quad (2)$$

**Table 1.** Summary of features for the TSP

Feature ID	Feature or feature type	Source
F1	Number of customers	[2, 18, 21]
F2	Area	[21]
F3, F4	Convex Hull (area and perimeter)	[21]
F5–F7	Smallest cluster (width, height, perimeter)	[21]
F8–F12	Distance related (general)	[2, 18, 21]
F13–F15	Angle related	[2]
F16–F21	Geographical variance	[18]
F22–F24	Proximity related	-
F25–F32	Polygon related	-

$$y = \cos(a_{lat}) * \sin(b_{lat}) - \sin(a_{lat}) * \cos(b_{lat}) * \cos(\Delta(a_{long}, b_{long})) \quad (3)$$

$$\beta_{a,b} = \arctan(x, y) \quad (4)$$

F13–F15 are based on latitude and longitude but can be converted to a Cartesian system without loss of generality by substituting the north-south line by one of the Cartesian axes. F16–F21 express geographical variance and dispersion by means of variance in latitude and longitude, variance of latitude multiplied with longitude, and the variance of the distance from the depot, cluster centroid, or cluster midpoint, to all customers in a route. Finally, we introduce two new types of features. First, F22–F24 count the number of customers within a certain radius from the depot, cluster centroid or cluster midpoint. These features have similar descriptive power as other already described features but might be more convenient to calculate. Second, F25–F32 are polygon related features. We split the smallest possible rectangle that can be fitted around all points into several equally-sized smaller rectangles, called polygons. Several features can be extracted from the polygon structure, such as the distance between the depot and the centroid of the polygon with the most customers, and the average distance from the cluster centroid to all activated polygons, i.e., polygons that contain customers. These features can capture the extent of concentration of customers at geographical locations.

**Model Extensions for the Vehicle Routing Problem.** For the VRP, we consider geographical data as well as demand data. The addition of demand data is imperative since the VRP involves multiple vehicle routes and vehicle capacity. The extension on the current model is summarized in Table 2. Here, F32–F36 describe the VRP instance considering the demand and vehicle capacity. F36 is similar to F35, but rounds up to the nearest integer. F38 is a feature for which we count the polygons in which the demand is higher than the average polygon-demand.

**Table 2.** Summary of additional features for the VRP

Feature ID	Feature	Source
F32	Total demand per instance	[18]
F33	Avg demand per instance	[18]
F34	Variance of demand per instance	[18]
F35	$\frac{\text{Total demand per instance}}{\text{Vehicle capacity}}$	[18]
F36	Minimum required vehicles	[21]
F37	$\frac{\text{Maximum customer demand in an instance}}{\text{Vehicle capacity}}$	[21]
F38	Polygon demand weight	-

### 3.3 Model Evaluation

In this section, we evaluate our generic TSP and VRP model using linear regression, random forests, and neural networks. The data is split into a training set and test set. The hyperparameters are tuned using 5-fold cross-validation grid search, with the Scikit-Learn Python library, after which two different automatic feature selection methods are employed. Examples of these hyperparameters are the tree-depth and the minimum number of samples required to split a node (random forest), the activation function,  $L_2$  penalty, hidden layer sizes, learning rate, and solver (neural network). Finally, the model is evaluated on the test set. In the next paragraph, the feature selection methods are briefly described. Next, the results of both the TSP and VRP model are presented.

**Feature Selection.** Feature selection is performed for several reasons. First, it indicates the individual importance of the features for the regression model. Second, features might be correlated, which potentially can distort some models. Also, a model can be overfitted because there are too many features relative to the available data. Finally, the computational costs need to be as low as possible for the approximation to be fast enough [21].

We employ two different methods for feature selection. The first method is called Elastic Net Regularization (ENR) [26]. 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 [26]. Ridge regression, however, does recognize grouped features but does not do automatic feature selection. ENR successfully combines these two methods. Nevertheless, 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. It should be noted that the features were standardized before fitting as this is necessary for coefficient shrinkage methods.



The second method employs recursive feature elimination, which can be used for random forests (RF-RFE). Random forests allow us to recursively remove features, based on feature importance. Since random forests is less affected by multicollinearity, we can assume that RF-RFE will render valuable results. For more details on both selection methods, we refer to [11] and [26]. 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.

**Model Performance.** We compare models using three different statistics: adjusted  $R^2$ , relative mean absolute error (rMAE) and relative root mean squared error (rRMSE). The adjusted  $R^2$  indicates the proportion of variance in the dataset that can be explained by the regression model; it 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 is useful for identifying large errors. Both MAE and RMSE are made relative, to the mean and standard deviation of the observed values, respectively.

$$\text{rMAE} = \frac{\frac{1}{N} \sum_{i=1}^N |\text{Predicted}_i - \text{Actual}_i|}{\text{Actual}} \quad (5)$$

$$\text{rRMSE} = \frac{\sqrt{\frac{\sum_{i=1}^N (\text{Predicted}_i - \text{Actual}_i)^2}{N}}}{\sigma_{\text{Actual}}} \quad (6)$$

Table 3 shows the performance of three models on the TSP dataset. We see that the adjusted  $R^2$  is high for all models. The relative MAE is low, with ENR performing the worst with a rMAE of 7% of the average distance in the data. The relative RMSE is also low, so it does not indicate large errors. ENR performs notably well in comparison with RF and NN. ENR eliminated 8 features and RF-RFE eliminated 6 features. ENR removed 2 out of the 3 proximity features (F22–F24), only the feature representing the proximity to the depot is kept. Some seemingly good features were removed because of redundancy. All polygon features (F25–F32) are in the model. RF-RFE made a different selection: it removed 6 of the 7 polygon features, with the only one remaining being the average number of customers in an activated polygon.

**Table 3.** Model performance for the TSP (5-fold cross-validation)

Performance	ENR	RF-RFE	NN
Number of Features	24	26	32
Adjusted $R^2$	0.938	0.966	0.965
rMAE	0.070	0.049	0.051
rRMSE	0.248	0.185	0.188

Table 4 shows the performance of the three models on the VRP dataset. Compared to the TSP model, the adjusted  $R^2$  data dropped significantly. Clearly, the distance becomes harder to predict when having multiple vehicles. Nevertheless, the performance is still acceptable, with a lower relative MAE but slightly larger errors compared to the TSP models. Again, linear regression performs relatively well compared to the more advanced methods. ENR removed several seemingly redundant features but kept all demand related features (F32–F38). RF-RFE now removed more features, which might be related to the noise in the data. For both the TSP and VRP model, we observe that the highest importance is given to the following features: number of customers (F1), convex hull area (F3), average distance between locations (F10), and the distance from the centroid of all activated polygons to the depot (F29).

**Table 4.** Model performance for the VRP (5-fold cross-validation)

Performance	ENR	RF-RFE	NN
Number of Features	27	23	38
Adjusted $R^2$	0.836	0.838	0.834
rMAE	0.048	0.047	0.047
rRMSE	0.404	0.402	0.406

### 3.4 Model Adaptations for the Waste Collection Planning

The waste collection problem and other IRPs differentiate 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 if there is too much overflow of containers. For our implemented benchmark method (Daganzo-approximation), we separately assess the service level requirements by adding an overflow penalty to the approximated distance. For our new approximation model, we can combine the performance indicators by both approximating the distance as the service level together.

We introduce two new features that are used to estimate the actual service level, namely the service level calculated using the expected fill levels (F39) and the average expected fill level of containers as a percentage of the container capacity (F40). For both features, we use the known container capacities to calculate the feature values. F39 can be calculated by considering for each container the days till last emptying, the average waste disposals per day for this container, and its capacity. We observe that F39 can estimate the service level reasonably well, when tested in a single-feature regression model aimed at predicting the service level we observe a mean absolute error of 9.3%. F40 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 closer to the capacity, the chance is higher it has overflowed. So, in case of equal expected service level and distance, the container with a higher fill level is favored.

After scaling both target variables on the domain  $[0, 1]$ , we define a new cost function (7), that combines distance and service level terms in a sequential objective function. The regression model estimates the costs, i.e., it is trained to predict the value of  $\zeta_{c,t}$ .

$$\zeta_{c,t}(S_t, x_{c,t}) = w^d * d_{S_t, x_{c,t}} + w^\alpha * \alpha_{S_t, x_{c,t}}, \quad \forall c \in C : C \subseteq I, \forall t \in \mathcal{T} \quad (7)$$

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  $t$ .  $x_{c,t}$  is the decision to insert container  $c$  on day  $t$ . The costs are determined using the predicted distance  $d$  and service level  $\alpha$ . The weights  $w$  strike a balance between the importance of the distance and the service level. In our experiments, we adjust both weights.

### 3.5 A 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. This framework can be applied to cases where the environment changes and the approximation model needs to be updated regularly, e.g., when the 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 next section, we will show how this framework can be applied to our case study.

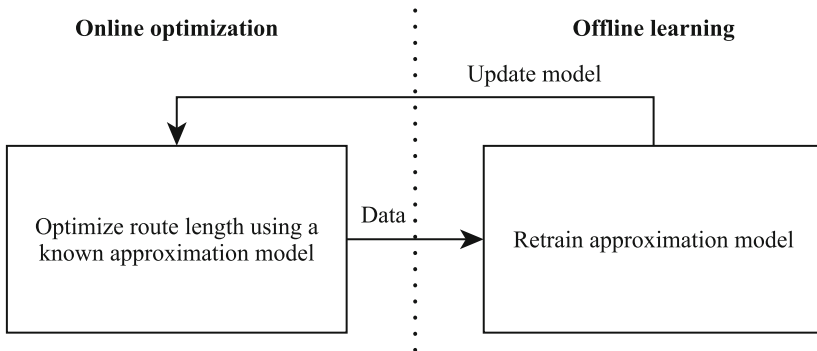


Fig. 2. Feedback loop with online optimization and offline learning

## 4 Computational Experiments and Results

To validate our proposed method, 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. 3). 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 3 replications of 125 days each with a 25-day warmup period.

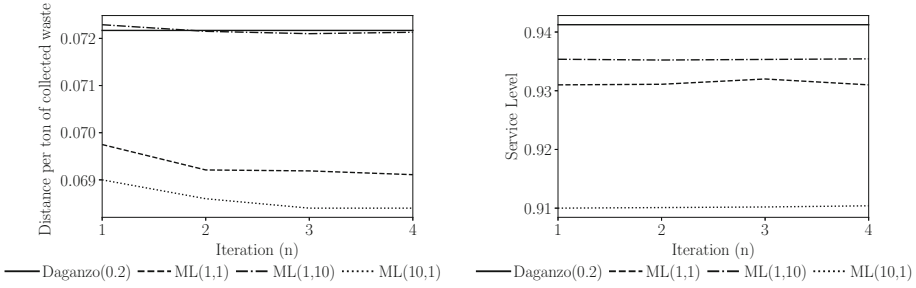
Three policies are compared: (i) the benchmark Daganzo-approximation with a service level penalty, (ii) our proposed machine learning model (ML), which combines distance and service level approximations, and (iii) a myopic policy that uses a horizon of  $T = 1$  and always favors the containers with the highest expected fill levels. For both the benchmark policy and our ML-model, the respective overflow penalty and approximation weights can be tuned. The tuning of these parameters can shift the focus, either favoring service level or distance. Table 5 summarizes the relevant experimental parameters for each model. The acceptable overflow penalty (AOP) is only used for Daganzo-approximation since the myopic method only considers container fill levels and our ML-model has its own service level approximation.

**Table 5.** Experimental parameters

Policy	Planning horizon	AOP	$(w^d, w^\alpha)$
Myopic	1	-	-
Daganzo	3	{0.1,0.2,0.3}	-
ML	3	-	{(1,1),(1,10),(10,1)}

To ease the presentation, we only show the results for the linear regression model, as its performance is close to those of the more advanced models, i.e., random forests or neural networks, see Sect. 3.3. 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 the 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 the method described in Sect. 3.5. The data used to obtain the initial approximation is generated using Daganzo-approximation. With the new model, we can expect the planning to change, to which we can adapt by means of the feedback loop. Figure 3 shows the respective distance and service level for the three settings of ML during several iterations of the feedback loop. The performance of the best setting for the Daganzo-approximation is also shown.



**Fig. 3.** Performance of approximation policies over several iterations, AOP = 0.2 (Daganzo) and  $(w^d, w^\alpha) = \{(1, 1), (1, 10), (10, 1)\}$  (ML),  $N = 4$

First, we see 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 6. First, the added value of a rolling planning horizon is confirmed by the bad performance of the myopic policy, in comparison with Daganzo and the regression model: the service level is relatively low, and the distance is over 15% more in comparison with the worst performing approximation method. Compared with the best performing approximation method, the myopic policy is more than 28% worse. Also, an additional vehicle is needed. Further, 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.

**Table 6.** Performance of approximation policies for all experiments, best ML-solution reported

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
ML (1,1)	0.0691	93.2%	4
ML (1,10)	0.0721	93.5%	4
ML (10,1)	0.0684	91%	4

## 5 Conclusions

We developed an approximation method, encompassing a large range of temporal and spatial features, which can be used to predict distances and service levels within transportation problems, to reduce the computational complexity of finding suitable vehicle routes. We illustrated the approach considering a large dynamic waste collection problem. As a benchmark, we implemented the Daganzo-approximation with a service level penalty. 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 that we can predict distance fairly accurately without solving the TSP or VRP, and explained the automatic feature selection methods for linear regression and random forests. 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 waste collection case with stochastic demands. The case study showed that fast approximation methods are valuable because they enable fast decision making. Our proposed model performs reasonably better than the benchmark policy.

Further research can be done on features that describe the problem instances more specifically. 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. Another research direction we propose is to use approximate dynamic programming within the customer selection phase, to iteratively learn a value function approximation.

## References

1. Archetti, C., Fernández, E., Huerta-Muñoz, D.L.: The flexible periodic vehicle routing problem. *Comput. Oper. Res.* **85**, 58–70 (2017)
2. Arnold, F., Sörensen, K.: What makes a VRP solution good? the generation of problem-specific knowledge for heuristics. *Comput. Oper. Res.* **106**, 280–288 (2019)
3. Baita, F., Ukovich, W., Pesenti, R., Favaretto, D.: Dynamic routing-and-inventory problems: a review. *Transport. Res. Part A: Policy Practice* **32**(8), 585–598 (1998)
4. Bard, J.F., Huang, L., Jaillet, P., Dror, M.: A decomposition approach to the inventory routing problem with satellite facilities. *Transport. Sci.* **32**(2), 189–203 (1998)
5. Beliën, J., De Boeck, L., Van Ackere, J.: Municipal solid waste collection and management problems: a literature review. *Transport. Sci.* **48**(1), 78–102 (2014)
6. Benjamin, A., Beasley, J.: Metaheuristics for the waste collection vehicle routing problem with time windows, driver rest period and multiple disposal facilities. *Comput. Oper. Res.* **37**(12), 2270–2280 (2010)
7. Braekers, K., Ramaekers, K., Van Nieuwenhuysse, I.: The vehicle routing problem: state of the art classification and review. *Comput. Ind. Eng.* **99**, 300–313 (2016)
8. Caceres-Cruz, J., Arias, P., Guimarans, D., Riera, D., Juan, A.A.: Rich vehicle routing problem, *Survey. ACM Comput. Surv.* **47**(2), 521 (2014)

9. Coelho, L.C., Cordeau, J.F., Laporte, G.: Thirty years of inventory routing. *Transport. Sci.* **48**(1), 1–19 (2014)
10. Dror, M., Laporte, G., Trudeau, P.: Vehicle routing with split deliveries. *Discrete Appl. Math.* **50**(3), 239–254 (1994)
11. Gregorutti, B., Michel, B., Saint-Pierre, P.: Correlation and variable importance in random forests. *Stat. Comput.* **27**, 13 (2013)
12. Gromicho, J., van Hoorn, J., Kok, A., Schutten, J.: Restricted dynamic programming: a flexible framework for solving realistic VRPS. *Comput. Oper. Res.* **39**(5), 902–909 (2012)
13. Heijnen, W.: Improving the waste collection planning of amsterdam, June 2019. <http://essay.utwente.nl/78290/>
14. Lalla-Ruiz, E., Voß, S.: A POPMUSIC approach for the multi-depot cumulative capacitated vehicle routing problem. *Optimization Lett.* **14**(3), 671–691 (2020)
15. Mes, M.: Using simulation to assess the opportunities of dynamic waste collection. In: Bangsow, S. (ed.) *Use Cases of Discrete Event Simulation: Appliance and Research*, chap. 13, pp. 277–307. Springer, Heidelberg (2012). [https://doi.org/10.1007/978-3-642-28777-0\\_13](https://doi.org/10.1007/978-3-642-28777-0_13)
16. Mes, M., Schutten, M., Rivera, A.P.: Inventory routing for dynamic waste collection. *Waste Manage.* **34**(9), 1564–1576 (2014)
17. Moin, N.H., Salhi, S.: Inventory routing problems: a logistical overview. *J. Oper. Res. Soc.* **58**(9), 1185–1194 (2007)
18. Nicola, D., Vetschera, R., Dragomir, A.: Total distance approximations for routing solutions. *Comput. Oper. Res.* **102**, 67–74 (2019)
19. Novoa, C., Storer, R.: An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. *Euro. J. Oper. Res.* **196**(2), 509–515 (2009)
20. Powell, W.B., Ryzhov, I.O.: *Optimal Learning and Approximate Dynamic Programming*, chap. 18, pp. 410–431. Wiley (2013)
21. Rasku, J., Kärkkäinen, T., Musliu, N.: Feature extractors for describing vehicle routing problem instances. In: Hardy, B., Qazi, A., Ravizza, S. (eds.) *5th Student Conference on Operational Research (SCOR 2016)*. OpenAccess Series in Informatics (OASISs), vol. 50, pp. 7:1–7:13. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2016)
22. Robust, F., Daganzo, C.F., Souleyrette, R.R.: Implementing vehicle routing models. *Transport. Res. Part B: Methodol.* **24**(4), 263–286 (1990)
23. Solomon, M.M.: Algorithms for the vehicle routing and scheduling problems with time window constraints. *Oper. Res.* **35**, 254–265 (1987)
24. Ulmer, M.W., Goodson, J.C., Mattfeld, D.C., Hennig, M.: Offline-online approximate dynamic programming for dynamic vehicle routing with stochastic requests. *Transport. Sci.* **53**(1), 185–202 (2019)
25. Ulmer, M.W., Mattfeld, D.C., Köster, F.: Budgeting time for dynamic vehicle routing with stochastic customer requests. *Transport. Sci.* **52**(1), 20–37 (2018)
26. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. *J. Royal Stat. Soc. Series B (Statistical Methodology)* **67**(2), 301–320 (2005)