





# A Scenario-Based Approach for the Vehicle Routing Problem with Roaming Delivery Locations under Stochastic Travel Times

Afonso Sampaio <sup>1</sup>, Joris Kinable <sup>1,2</sup>, Lucas P. Veelenturf <sup>1</sup>, and Tom Van Woensel <sup>1</sup>

<sup>1</sup>Eindhoven University of Technology, NL

<sup>2</sup>Robotics Institute, Carnegie Mellon University, USA

May 7, 2019

## Abstract

We address a stochastic variant of the Vehicle Routing Problem with Roaming Delivery Locations. In this model, direct-to-consumer deliveries can be made in the trunk of the customer’s car, while the vehicle is parked at a location along the customer’s itinerary. The stochasticity arises from the uncertainty in travel times and the problem is formulated as a two-stage stochastic model. We propose a scenario-based sample average approximation to obtain a heuristic solution. Several experiments to assess the effect of our solution approach compared to a pure deterministic solution approach using expected travel times, show that a cost savings of on average more than 30% can be obtained. Furthermore, it is shown that the flexibility provided by using alternative roaming delivery locations as a recourse to avoid missed deliveries can provide, on average, costs savings of 25% compared to a recourse staying with the locations chosen in the a priori first stage plan.

**Keywords**— trunk-delivery, stochastic travel times, sample average approximation

## 1 Introduction

Private cars represent a large share of the total flow of vehicles within urban areas. Given current population and urbanization growth, they will likely continue to have a prominent role in the fulfillment of transportation activities for years to come. However, some studies reveal that the average car is parked most of the time and only being used for a relatively short period – parked at home for 80% of the time, 16.5% elsewhere and moving for 3.5% (Bates and Leibling, 2012). Improving the period of time when the vehicle is actually moving e.g., minimize travelled distance to reduce costs/emissions, was the focus of attention for most academic studies. However, a comprehensive view seeking for more cost-effective, sustainable solutions to minimize flows of goods and people needs to leverage all available resources at all times.

Trunk delivery refers to using the trunk of a customer’s car, when the vehicle is parked, as a possible location for the final delivery of goods to the consumer. As indicated, cars are parked away from home for a significant amount of time, thus providing opportunities that could be seized to avoid, for example, missed deliveries and, consequently, multiple visits at home. In fact, given the explosion of e-commerce and on-line shopping, the last-mile supply chains of even the largest e-tailers are strained by the sheer volume increase of direct-to-consumer orders. This challenge is even amplified by the increased customer service levels offered by e-tailers to compete against the instant gratification of brick-and-mortar stores. Companies are evaluating new and innovative business models, such as trunk delivery, that could help improving last-mile operations. As an example, Amazon recently introduced a free in-car delivery service for some of its customers. In a partnership with a selection of car manufacturers, the service connects to the vehicle allowing the Amazon delivery person an one-time access to the trunk of the customer’s car. Whereas neither manufacturers nor Amazon expects monetary gains from the service, they see it as an added delivery option to market (Hawkins, 2018).

In the literature, the Vehicle Routing Problem with Roaming Delivery Locations (VRPRDL) is introduced by Reyes et al. (2017). In the VRPRDL, each customer has a given, fixed, itinerary specifying one or more locations where his/her car will be parked, within corresponding time windows i.e. a roaming delivery location. While this work showed the benefits of trunk delivery, specially combined with traditional home delivery, a fully deterministic environment is assumed. Nevertheless, companies may face many uncertain disruptions that could affect travel times e.g., accidents, vehicle breakdowns, weather and, consequently, hinder the fulfillment of a delivery within the time windows during which the customer’s car will be parked at a given location. Moreover, since a customer is also moving from one location to another along his/her itinerary, the time during which the customer’s vehicle will be available for trunk delivery at a given location might also be delayed due to increased travel times. The VRPRDL and stochastic travels times is also addressed by Lombard et al. (2018). The authors propose a Monte-Carlo method in which many deterministic VRPRDL instances, for a given travel time realization which is sampled out of the considered travel time distribution, are solved until the variation of the costs found is marginal. After a simulation, a frequency analysis is performed to assess the (probability of) occurrence of the best solutions and the costs of the most frequent solutions. However, this approach is more appropriate for a *wait-and-see* context: the problem is solved for the revealed travel times and then simulated to obtain the quality of the solution. For other contexts, where the routing plan has to be decided before travel times are known, such an approach yields very poor solutions, since the routing plan obtained is optimized towards a particular travel time matrix, which might not necessarily realize.

In this work, we consider the VRPRDL and Stochastic Travel Times (VRPRDL-STT). We propose tackling the problem as a two-stage, stochastic optimization problem. Conceptually, in the first stage, we obtain the *a priori* VRPRDL routing plan (also known as first-stage decision) which abides to the time windows of any location visited, considering deterministic travel times values. In the second stage, travel times are revealed and the *a priori* plan is modified by a given recourse policy whenever a failure, i.e. a late visit to a customer’s location, occurs. The objective is to obtain an *a priori* routing plan minimizing both the expected travel costs (routing) and the costs incurred in the second stage following the recourse policy. For many different types of Vehicle Routing Problems with Time Windows (VRPTW), servicing a location outside its associated time windows is acceptable, to some extent, although incurring in waiting time in case of early arrivals and a penalty in case of late arrivals. For the VRPRDL,

the delivery locations, the trunk of a car, are only present within a given time windows. Waiting is allowed, but late arrivals are not feasible, since the car will not be present anymore at the location. We propose different recourse policies to recover feasibility in the a priori plan given realized travel times in the second-stage. For example, if due to the realized travel times a particular customer location cannot be visited within its corresponding time windows, another location in the customer itinerary might be selected.

To solve the two-stage problem, we propose a solution framework which combines an extension of the local search heuristic devised by Reyes et al. (2017), to account for uncertain travel times, with an implementation of the Sample Average Approximation (SAA) approach proposed by Kleywegt et al. (2002). We assume that travel times follow a known probability distribution, allowing for the generation of (a very large number of) different possible *scenarios*. The SAA method is an iterative procedure that solves the problem restricted to a small set of scenarios on each iteration and evaluates the obtained solution over a (considerable) larger set of (independent) scenarios to approximate the expected objective function value. We note that, even if the chosen probability distribution of the travel time follows the convolution property, integrating the evaluation of the recourse policies proposed in this work in an exact approach is challenging. The SAA method is suitable to deal with extremely large scenarios sets, thus able to approximate the optimal value of the stochastic problem. Moreover, exact and heuristic solution approaches can be applied to solve each restricted problem in a SAA iteration. In this paper, we adapted the local search algorithm by Reyes et al. (2017) to include the computation of the recourse cost for a number of recourse policies.

The contributions of this research are:

- We consider a last-mile application, considering *trunk delivery*, taking into account stochastic travel times. We present a two-stage stochastic formulation with recourse to model the problem. We propose a scenario-based stochastic approximation (SAA) method to solve the problem combined with an adaptation of the local search heuristics proposed in Reyes et al. (2017) considering recourse costs to evaluate neighborhood solutions. In our computational experiments, we explicitly evaluate implementation decisions taken into the design of the SAA method (choice of a sample size to estimate better lower bounds, trade-off between solution quality and computational time, number of iterations).
- We exploit the flexibility introduced by trunk-delivery i.e., a delivery can be made at different locations at different times. Specifically, we define a variety of recourse policies reducing the number of failed deliveries caused by travel time stochastics.
- We show the benefits of using a stochastic solution approach over the deterministic counterpart are twofold. First, the evaluation of different travel time scenarios in the a priori stage allows for a better assessment of the customer availability and to hedge against the travel time uncertainty. Results show that, compared to a deterministic solution approach, which takes average travel times as input, the proposed methodology provides savings of, on average, more than 30%. Secondly, exploiting the flexibility to visit customers at other locations than planned for in the a priori plan, leads on average to cost savings of 18%.

The remainder of the paper is organized as follows. Section 2 reviews some of the works related to the VRPRDL as well as issues regarding the consideration of stochastic elements in vehicle routing problems, in particular stochastic travel times. The mathematical notation

used throughout the text is introduced in Section 3, as well as a formal problem definition and the mathematical programming formulation proposed by Reyes et al. (2017), considering deterministic travel times. We extend the formulation to a two-stage stochastic model to account for uncertainty in travel times. In Section 4, we describe our implementation of a scenario-based Sample Average Approximation. Section 5 presents computational experiments carried to evaluate the proposed methodology and to assess the gains of considering uncertainty in travel times when planning the schedules of drivers in the context of trunk delivery. Finally, our conclusions and directions for further research are presented in Section 6.

## 2 Related Literature

The VRPRDL was first introduced in Reyes et al. (2017). The authors developed construction and improvement heuristics and assessed the benefits of trunk delivery, showing that a reduction of up to 50% on total travelled distance can be achieved for certain environments. A followup work by Ozbaygin et al. (2017) was the first to tackle the problem with an exact approach, introducing a Branch-and-Price method for the VRPRDL able to solve instances of up to 120 customers. In both works, all relevant information (travel times, customers, demands) are assumed to be deterministic.

The problem relates to both the VRPTW and to the Generalized Vehicle Routing Problem (GVRP). For the former, the reader is referred to Toth and Vigo (2014) for an extensive coverage of many problem variants and state-of-the-art methods. The latter was introduced by Ghiani and Improta (2000) and consists of a generalization of the VRP for which the set of customers is partitioned in clusters and exactly one customer of each cluster has to be visited in the solution. The reader is referred to Bektaş et al. (2011) for formulations and an exact method to solve the GVRP and to Kovacs et al. (2015) for an application of the problem and a heuristic solution approach. Moccia et al. (2012) introduced the GVRP with Time Windows (GVRPTW) and proposed a tabu search heuristic to solve the problem. Observe that the GVRPTW reduces to the VRPRDL when all the locations of a given cluster define the customer’s itinerary and the associated time windows of each location within a cluster are non-overlapping.

Recently, increased attention has been given to numerous stochastic variants of the VRP(TW). The uncertain events addressed in most of the works comprise customers availability, when the presence or absence of a customer is a random event; demand volumes, when the exact amount of a commodity to collect or deliver is unknown; operation times, when travelling or service times, for example, are considered stochastic. Gendreau et al. (2016) provide an overview of the state-of-the-art for those main classes of stochastic VRPs, the modelling issues and the exact and approximate methods that have been proposed to solve them. The uncertainty on travel times addressed in this work stem mainly from unexpected disruptions affecting expected values e.g., increased traffic due to accidents or weather conditions. Expected variations on travel times such as due to congestion during peak-hours (e.g., early morning and late afternoon rush) can be tackled using deterministic models by considering time-dependent travel times. The reader is referred to Gendreau et al. (2015) for an overview of such problems.

Usually, a VRPTW with stochastic travel times (VRPTW-STT) takes into account deviations from customer’s time windows in the computation of the expected recourse costs. In problems with soft time windows, late arrivals are allowed in the routing plan, but usually incurring in a penalty proportionally to their tardiness and a recourse action is not necessary. As observed by Gendreau et al. (2016), this allow for the use of closed-form expressions to compute

the expected total penalty of a route – i.e., the expected earliness and tardiness – if the probability distribution follows the convolution property. A shifted gamma distribution (Erlang) is considered by Russell and Urban (2008). The authors propose different functions to evaluate penalty costs (fixed, linear, quadratic) and a tabu-search heuristic to solve the problem. Li et al. (2010) considered both stochastic travel and service times modeled by a normal distribution with parameters (mean, variance) depending on the arcs and customers, respectively. The authors propose a chance constrained programming and a stochastic programming model and solve both models with an adaptation of a tabu-search heuristic. Soft time windows are also considered by Taş et al. (2013). The authors assume the travel time of one unit of distance as a random variable with gamma distribution by given parameters (scale, shape). The travel time for an arc is then obtained by scaling the unit travel time with respect to the arc distance. A tabu-search heuristic is proposed to solve the problem and the authors consider different coefficients of variation for the travel time per unit distance to assess solutions obtained regarding variability. The same problem is solved exactly through a branch-and-price method by Taş et al. (2014b) and under time-dependency of the travel times by Taş et al. (2014a). Deadlines for visiting (a subset of) customers under stochastic travel times are considered by Adulyasak and Jaillet (2016). The problem is extended to consider soft time windows, and a branch-and-cut framework is proposed.

For problems with hard time windows, such as the VRPRDL(STT), late arrivals at customers are not allowed, requiring a recourse action to be applied to recover feasibility. Modelling expected arrival times at customers cannot be done, generally, by applying convolution properties of the distribution used to model travel times, as hard windows tend to truncate the distribution. Only a few works have addressed problems with hard-time windows and stochastic times and, due to the difficulty in computing expected times, the recourse policies proposed are usually restrictive e.g., assume that only one disruption occurs in a route, and only one recourse action is necessary (Errico et al., 2016). A VRPTW with stochastic travel times and stochastic demands is considered by Branda (2012), where the problem was modelled as a stochastic programming problem with chance constraints and a sample average approximation technique was used to derive estimates on the sample sizes required to obtain a feasible solution. Normal distributed travel times are considered by Ehmke et al. (2015) and a chance-constrained model is proposed in order to guarantee a given service level for customers (a probability on respecting the time windows). Those features can be embedded in any algorithm for the VRPTW with stochastic travel times and demands, and the authors show how to apply them on a tabu search heuristic. Binart et al. (2016) address a variation of the stochastic VRPTW, considering both stochastic travel and service times, in which customers are split in mandatory and optional. The first have to be served within their time windows whereas the latter can be serviced at any time during the planning horizon or not be serviced at all. The objective is to minimize total travel time servicing as many optional customers as possible. Stochastic travel and service times are dealt through dynamic programming where optional customers are used as buffers to hedge against the variations on those elements. Finally, Miranda and Conceição (2016) propose a statistical model to compute the cumulative probability function for the arrival times over customers when travel times are normally distributed.

Only a few works addressing stochastic events in the context of the GVRP(TW) exist. Similarly to the work by Laporte et al. (2002) on the VRP with stochastic demands, Biesinger et al. (2016) consider a recourse policy which consists of returning trips to the depot whenever a failure – a stock-out – occurs and propose an Integer  $L$ -Shaped Method (Laporte and Louveaux,

1993) and branch-and-cut framework to solve a GVRP with stochastic demands.

### 3 Problem Description

We first provide a general description of the VRPRDL and the MIP formulation proposed by Reyes et al. (2017), adapted to our notation, in Section 3.1 and 3.2, respectively. In Section 3.3, we consider that travel times are random variables and describe a two-stage stochastic formulation for the VRPRDL with Stochastic Travel Times (VRPDL-STT).

#### 3.1 The Vehicle Routing Problem with Roaming Delivery Locations

In the VRPRDL, a set of customers  $C = \{1, 2, \dots, n\}$  is serviced by a homogeneous fleet of vehicles having a fixed capacity  $Q$  operating during the planning period  $[0, T]$ . Each customer  $c \in C$  has a non-negative demand  $d_c$  and is associated with an unique set of locations  $N_c$ , specifying the itinerary of customer  $c$  i.e., the locations and times at which his/her car is available for trunk delivery. Assuming  $N_c = \{i_1^c, i_2^c, \dots, i_{|N_c|}^c\}$  and that  $i_1^c = i_{|N_c|}^c$  represent the home location for any customer  $c$ , his/her itinerary is defined by the directed graph  $G(N_c, A_c)$ , where  $A_c = \{(i_1^c, i_2^c), (i_2^c, i_3^c), \dots, (i_{|N_c|-1}^c, i_{|N_c|}^c)\}$ . A delivery vehicle servicing a customer  $c \in C$  visits exactly one of the delivery locations in  $N_c$  i.e., does not traverse arcs in  $A_c = \bigcup_{c \in C} A_c$ . Let

$N = (\bigcup_{c \in C} N_c) \cup \{0, n+1\}$  be the set of *service locations*, where 0 is the source depot and  $n+1$  the target depot of the vehicles. The route of a vehicle is defined over the directed graph  $G(N, A_R)$ , where  $A_R = (\bigcup_{\substack{c_1, c_2 \in C \\ c_1 \neq c_2}} (N_{c_1} \times N_{c_2})) \cup (\bigcup_{c \in C} (\{0\} \times N_c)) \cup (\bigcup_{c \in C} (N_c \times \{n+1\}))$ .

A non-negative travel time,  $t_{ij}$ , and cost  $w_{ij}$  (e.g. distance) for traversing the arc are associated with each arc  $(i, j) \in A = A_c \cup A_R$ , and arc's travel times are the same for customers and delivery vehicles. Servicing customer  $c \in C$  can occur at any of the locations  $i \in N_c$ , but only during a location-dependent time-interval  $[a_i, b_i]$ . In particular, the time windows for any customer  $c \in C$  are non-overlapping, defining an ordering on the location set  $N_c$ , and are defined as:

$$a_1^c = 0; \quad b_{|N_c|}^c = T \quad (1)$$

$$a_l^c = b_{l-1}^c + t_{i_{l-1}^c, i_l^c} \quad \forall l = 2, \dots, |N_c| \quad (2)$$

that is, a customer starts at home and moves from one location to another in his/her itinerary throughout the planning period  $[0, T]$ , being available for trunk delivery at location  $i^c \in N_c$  when the car is parked (not moving), between times  $a_i$  and  $b_i$ . It is assumed that, for a pair of distinct customers  $c, c' \in C$ ,  $N_c \cap N_{c'} = \emptyset$ , i.e., customers do not share locations. This assumption is readily satisfied by introducing duplicate locations when necessary. Since there exists an unique correspondence between a location and a customer, we define the function  $\pi : N \setminus \{0, n+1\} \mapsto C$  to map a location  $i \in N$  to the unique customer it belongs to, i.e.  $i \in N_{\pi(i)}$ . All routes start at the source depot, 0, and end at the target depot,  $n+1$ , within the planning period i.e.,  $a_0 = a_{n+1} = 0$  and  $b_0 = b_{n+1} = T$ , and should not exceed vehicle capacity. A solution to the VRPRDL constitutes a set of routes in which each customer,  $c \in C$ , is serviced exactly once within the time windows of the selected location for service,  $i \in N_c$ . The objective is to find a set of routes minimizing the total costs e.g., distance travelled. By definition,

the problem generalizes to the well-known Generalized VRP with time window constraints. Moreover, when  $|N_c| = 1$  for all  $c \in C$ , the problem reduces to a standard Capacitated VRP with Time-Windows. It follows that the VRPRDL belongs to the class of NP-Hard problems. Throughout this paper, the following notation is used. Given a subset of vertices  $S \subseteq V$ , the cutset  $\delta(S)$  denotes the set of edges with exactly one endpoint in  $S$ . The cutset  $\delta^+(S)$  denotes the set of directed edges (arcs) having their tail in  $S$  and their head not in  $S$ . Similarly, the cutset  $\delta^-(S)$  denotes the set of edges with their head in  $S$  and their tail outside  $S$ . For undirected edges  $\delta(S) = \delta^+(S) = \delta^-(S)$ . A summary of the notation used throughout this paper is given in Table 1.

Parameter	Description
$C$	Set of customers
$N_c$	Set of locations for customer $c \in C$
$Q$	Vehicle capacity
$d_c$	Demand of customer $c \in C$
$t_{ij}$	Travel time from $i$ to $j$
$w_{ij}$	Cost to travel from $i$ to $j$
$[a_i, b_i]$	Time window associated with location $i \in N$

Table 1: Mathematical notation used throughout the text

### 3.2 Deterministic MIP Model

To unambiguously define the problem, we re-state the MIP model proposed by Reyes et al. (2017), adapted to our notation. The model uses binary routing variables  $x_{ij}$ , indicating whether a delivery vehicle traverses arc  $(i, j) \in A_R$ , continuous start time variables  $\tau_c$  to record the service start time of customer  $c \in C$ , and continuous load variables  $y_c$  to track the vehicle capacity remaining after visiting customer  $c \in C$ .

$$\min \sum_{(i,j) \in A_R} w_{ij} x_{ij} \quad (3)$$

$$\text{s.t.} \quad \sum_{(i,j) \in \delta^+(N_c)} x_{ij} = 1 \quad \forall c \in C \quad (4)$$

$$\sum_{(j,i) \in \delta^-(N_{\pi(i)})} x_{ji} = \sum_{(i,j) \in \delta^+(N_{\pi(i)})} x_{ij} \quad \forall i \in N \setminus \{0, n+1\} \quad (5)$$

$$y_c \geq y_{c'} + d_{c'} - Q(1 - \sum_{i \in N_c} \sum_{j \in N_{c'}} x_{ij}) \quad \forall c \in C \cup \{0\}, c' \in C \setminus \{c\} \quad (6)$$

$$\tau_{c'} \geq \tau_c + \sum_{i \in N_c} \sum_{j \in N_{c'}} t_{ij} x_{ij} - T(1 - \sum_{i \in N_c} \sum_{j \in N_{c'}} x_{ij}) \quad \forall c \in C \cup \{0\}, c' \in C \setminus \{c\} \quad (7)$$

$$\sum_{i \in N_c} a_i \sum_{(i,j) \in \delta^+(N_c)} x_{ij} \leq \tau_c \leq \sum_{i \in N_c} b_i \sum_{(i,j) \in \delta^+(N_c)} x_{ij} \quad \forall c \in C \quad (8)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A_R \quad (9)$$

$$0 \leq y_c \leq Q - d_c \quad \forall c \in C \quad (10)$$

$$0 \leq \tau_c \leq T \quad \forall c \in C \quad (11)$$

The objective function minimizes the total travel cost for the vehicles. Constraints (4) ensure that all customers are serviced and constraints (5) ensure that flow conservation is preserved over all locations. Capacity constraints are imposed by inequalities (6): the sum of customer demands served in a single trip cannot exceed the vehicle capacity. Inequalities (7) guarantees that the correct amount of driving time is spent between servicing customers  $c$  and  $c'$ . Finally, service within one of the availability time windows of the customer is imposed by (8). A vehicle is allowed to wait if it arrives early at a customer location, but it cannot be late.

### 3.3 A Two-Stage Stochastic Model

Conceptually, consider that planning decisions are taken in two stages. In the first stage, an a priori routing plan, i.e. a set of a priori routes each starting and ending at the depot and all customers assigned to exactly one route, is computed using the deterministic values for the arc's travel times. In the a priori plan, all deliveries to customers occur within the corresponding time windows of the locations and vehicles return to the depot within the time horizon. Obviously, travel time disruptions might occur. In particular, we consider disruptions increasing travel time between two locations. Consequently, customers might not be serviced within their corresponding time windows following the schedule of the a priori plan and the delivery vehicle might incur overtime. In this case, recourse actions are taken (and possibly costs are incurred) to modify the routes such that customers are serviced within their windows and vehicles do not exceed a maximum overtime period.

For example, one possible recourse action (in the next section, we elaborate more on all recourse actions considered in this paper) could be to delivery to a different location within the customer's itinerary. Note, however, that a customer's vehicle moves from one location to another in his/her itinerary and is subjected to the same disruptions in travel times experienced by the delivery vehicles. In particular, the extend of a disruption along the arcs in the itinerary,  $A_c$ , of a customer  $c \in C$  might change the availability for service in the locations  $i \in N_c$  i.e., change the time windows  $[a_i, b_i]$ . Let  $\delta_{ij} \geq 0$  be the length of the disruption in arc  $(i, j) \in A$ ,  $\tilde{\omega}_{ij} = t_{ij} + \delta_{ij}$  the realized travel time during the second-stage and  $[\tilde{a}_i, \tilde{b}_i]$  the time window for location  $i$  after disruptions realize. Figure 1 illustrates the itinerary of a customer  $c \in C$ , with four locations  $N_c = \{i_1^c, i_2^c, i_3^c, i_4^c\}$ .

After disruptions are revealed (in the second stage) and travel times changed, the customer arrives at location  $i \in N_c$  at  $\tilde{a}_i \geq a_i$ , given the disruptions in his/her itinerary. We assume that customer's latest time at location  $i$ ,  $b_i$ , does not change. In particular, if  $\tilde{a}_i > b_i$ , then location  $i \in N_c$  is no longer available for servicing customer  $c$ . In the example, location  $i_2$  is not available for delivery anymore in the second stage. The time window for location  $i_1$  remains the same and the time windows for locations  $i_3$  and  $i_4$  are shortened.

We now extend the the MIP formulation (3) – (11) to include stochastic travel times. First, let  $\omega = (\omega_{ij})_{(i,j) \in A}$  be a random variable vector, where  $\omega_{ij}$  is the stochastic travel time for traversing arc  $(i, j)$ , and  $\tilde{\omega} = (\tilde{\omega}_{ij})_{(i,j) \in A}$  a particular realization of  $\omega$ . To represent recourse actions taken after the realization of the travel times (second-stage variables), we use similar variables as proposed by Hvattum et al. (2006) for a VRP with stochastic customers. Let  $x_{ij}^+, (i, j) \in A_R$ , be a binary variable indicating whether ( $x_{ij}^+ = 1$ ) a vehicle traverses arc  $(i, j)$  in the second-stage but not in the first-stage solution  $\mathbf{x} = (x_{ij})_{(i,j) \in A_R}$ . Similarly, let  $x_{ij}^-, (i, j) \in A_R$



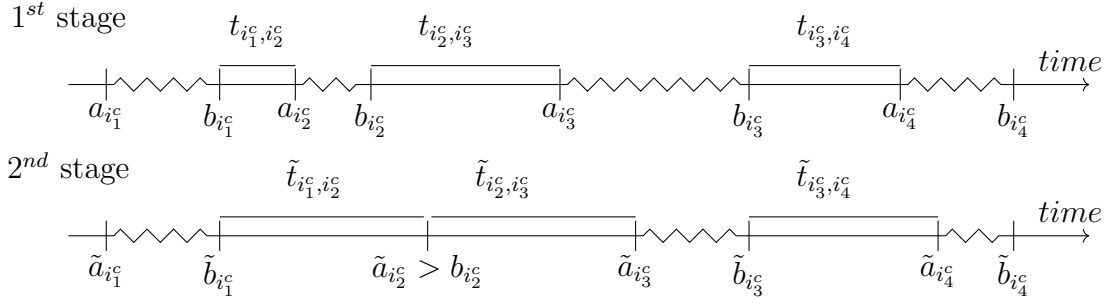


Figure 1: Changes in location's time windows for a particular travel time scenario.  $\tilde{\omega}_{i,j} = t_{i,j} + \delta_{i,j}$ .

be a binary variable indicating whether ( $x_{ij}^- = 1$ ) a vehicle traverses arc  $(i, j)$  in the first-stage but not in the second-stage solution, after recourse. The continuous variable  $\tau_c^+$  track the (potentially new) service time of customer  $c \in C$  after recourse. We now let  $x_{ij}^+$ ,  $x_{ij}^-$  and  $\tau_c^+$  all depend on the random event  $\omega$  to emphasize that those decisions variables are defined for each travel time realization vector  $\tilde{\omega}$ . Similarly, let  $[a_i(\omega), b_i(\omega)]$  denote the time windows for location  $i \in N$  as a function of the random travel time variable  $\omega$  i.e.,  $[\tilde{a}_i, \tilde{b}_i]$  is the time windows for a realized vector  $\tilde{\omega}$ .

The structure used to define the recourse actions in the second-stage is said to be relatively complete if for every first-stage solution and every possible realization of random data, the second-stage problem is feasible. Observe that the structure  $(\mathbf{x}^+, \mathbf{x}^-)$  does not provide relatively complete recourse, as some customers might not be serviced in time at any of his/her locations depending on the realized arc travel times. However, a model with relatively complete recourse can be achieved if skipping service for some customers is allowed, incurring in a high penalty (Hvattum et al., 2006). Let  $\lambda_c(\omega)$  be a binary variable indicating whether customer  $c \in C$  is skipped ( $\lambda_c(\omega) = 1$ ) in the second stage solution, and  $\Lambda_c$  the penalty for skipping customer  $c$ . When vehicles return to the depot, each unit of time exceeding the time horizon,  $T$ , incurs in a penalty cost of  $\beta$ . Moreover, let binary variables  $u_{c,c'}$  indicate whether customers  $c, c' \in C$  are serviced by the same first-stage route.

$$\min \sum_{(i,j) \in A} w_{ij} x_{ij} + \mathbb{E}[R(\mathbf{x}, \mathbf{x}^+, \mathbf{x}^-, \tau^+, \lambda, \omega)] \quad (12)$$

$$\text{s.t.} \quad (4) - (11)$$

$$\sum_{(i,j) \in \delta^+(N_c)} (x_{ij} + x_{ij}^+(\omega) - x_{ij}^-(\omega)) + \lambda_c(\omega) = 1 \quad \forall c \in C \quad (13)$$

$$\sum_{(j,i) \in \delta^-(N_{\pi(i)})} (x_{ij} + x_{ij}^+(\omega) - x_{ij}^-(\omega)) = \sum_{(i,j) \in \delta^+(N_{\pi(i)})} (x_{ij} + x_{ij}^+(\omega) - x_{ij}^-(\omega)) \quad \forall i \in N \setminus \{0, n+1\} \quad (14)$$

$$\tau_{c'}^+(\omega) \geq \tau_c^+(\omega) + \sum_{i \in N_c} \sum_{j \in N_{c'}} \omega_{ij} (x_{ij} + x_{ij}^+(\omega) - x_{ij}^-(\omega)) -$$

$$T(1 - \sum_{i \in N_c} \sum_{j \in N_{c'}} (x_{ij} + x_{ij}^+(\omega) - x_{ij}^-(\omega))) \quad \forall c \in C \cup \{0\}, c' \in C \setminus \{c\} \quad (15)$$

$$\tau_c^+(\omega) \geq \sum_{i \in N_c} a_i(\omega) \sum_{(i,j) \in \delta^+(N_{\pi(i)})} (x_{ij} + x_{ij}^+(\omega) - x_{ij}^-(\omega)) \quad \forall c \in C \quad (16)$$

$$\tau_c^+(\omega) \leq \sum_{i \in N_c} b_i(\omega) \sum_{(i,j) \in \delta^+(N_{\pi(i)})} (x_{ij} + x_{ij}^+(\omega) - x_{ij}^-(\omega)) \quad \forall c \in C \quad (17)$$

$$x_{ij}^+(\omega) + x_{ij}^-(\omega) \leq 1 \quad \forall (i,j) \in A_R \quad (18)$$

$$c, c' \text{ on the same first-stage route} \iff u_{c,c'} = 1 \quad \forall c, c' \in C \quad (19)$$

$$x_{ij}^+(\omega) \leq u_{\pi(i), \pi(j)} \quad \forall (i,j) \in A_R, i \neq 0, j \neq n+1 \quad (20)$$

$$x_{ij}^+(\omega) \in \{0, 1\} \quad \forall (i,j) \in A_R \quad (21)$$

$$x_{ij}^-(\omega) \in \{0, 1\} \quad \forall (i,j) \in A_R \quad (22)$$

$$0 \leq \tau_c^+(\omega) \leq T \quad \forall c \in C \quad (23)$$

$$\lambda_c(\omega) \in \{0, 1\} \quad \forall c \in C \quad (24)$$

$$u_{c,c'} \in \{0, 1\} \quad \forall c, c' \in C \quad (25)$$

where

$$R(\mathbf{x}, \mathbf{x}^+, \mathbf{x}^-, \boldsymbol{\tau}^+, \boldsymbol{\lambda}, \omega) = \sum_{(i,j) \in A_R} w_{ij}(x_{ij}^+(\omega) - x_{ij}^-(\omega)) + \sum_{c \in C} \Lambda_c \lambda_c(\omega) + \beta \sum_{c \in C} \sum_{i \in N_c} \max\{0, \tau_c^+(\omega) + \omega_{i0} - T(1 - \sum_{i \in N_c} (x_{i0} + x_{i0}^+(\omega) - x_{i0}^-(\omega))) - T\}$$

The objective is to minimize total routing cost for the first stage solution and expected (random) recourse costs  $R(\mathbf{x}, \mathbf{x}^+, \mathbf{x}^-, \boldsymbol{\tau}^+, \omega)$ , encompassing the cost of changes required in the first stage routing and penalties for skipping customers and overtime after recourse. Constraints (13)-(14) ensure that customers are either visited or skipped in the second stage solution. Cost  $\Lambda_c$ ,  $c \in C$ , should be set to a high value, such that skipping customer  $c$  in the second-stage only occurs when it is not feasible to service  $c$  within its corresponding time windows. Observe that time windows for customer locations in the second stage,  $a_i(\omega)$  and  $b_i(\omega)$ , depend on the realization of the random vector  $\omega$  and are, thus, random variables as well. Inequalities (15)-(17) ensure that service times for not skipped customers in the second stage occur within their corresponding time windows,  $\tilde{a}_i$  and  $\tilde{b}_i$ , given the revealed travel times  $\tilde{\omega}$ . Inequalities (18) are added to make the definitions of second stage variables  $x_{ij}^+(\omega)$  and  $x_{ij}^-(\omega)$  consistent, but are not necessary since they are only used together in the form  $(x_{ij}^+(\omega) - x_{ij}^-(\omega))$ . Note that, if arc  $(i, j) \in A_R$  is traversed both in the first and second stage, then  $x_{ij}(\omega) = 1$  and  $x_{ij}^+(\omega) = x_{ij}^-(\omega) = 0$  (or, also,  $x_{ij}^+(\omega) = x_{ij}^-(\omega) = 1$ , if inequalities (18) are not enforced). Inequalities (19)-(20) impose that a second stage route services only customers assigned to the same first stage route. Such restriction could be, for example, due to sorting and bundling processes being done considering the first stage solution and which are not easily or quickly changed after recourse. Thus, a complete rescheduling is not allowed – reassignment of customers to a different vehicle route or the start of new routes is not allowed after recourse – and we only allow that customers may be rearranged within their corresponding first stage route and that a different location from the one chosen to service a customer in the first-stage may be selected

to service that customer after recourse. Consequently, vehicle capacity does not need to be enforced in second-stage constraints. For the sake of simplicity, inequalities (19) are stated as informal implications in the model but can be formulated as linear constraints, as shown in the Appendix.

In formulating the two-stage model, we assume that the random variable  $\omega$  has finite support. We have one recourse structure  $(\mathbf{x}^+, \mathbf{x}^-, \tau^+, \lambda)$  and constraints (13)-(25) defined for each possible realization  $\tilde{\omega}$ . Even if the particular distribution used to model travel times is not discrete, we can approximate the optimal value of the problem by considering  $\omega$  taking values over a discrete, finite subset of its support i.e., we consider a (potentially large) set of realizations of the distribution. Another difficulty with the formulation is that the recourse actions allowed by the model are less restrictive than those usually considered in the literature, specially regarding hard-time windows. For example, Errico et al. (2016) consider that a route needs at most one recourse action. More intricate actions makes evaluating the recourse cost more complex and, even when simpler actions are considered, computing the value  $R$  with integer recourse (in the case  $(\mathbf{x}^+, \mathbf{x}^-)$ ), requires solving many similar integer NP-Hard problems (Schultz et al., 1998).

Not surprisingly, it is usually the case that solving two-stage stochastic VRPs to optimality is only possible, from a computational point of view, for problems with few customers. Thus, we resort to a heuristic scenario-based method to solve the VRPRDL-STT considering fixed recourse actions applied to a first-stage solution to compensate for the possible infeasibilities arising after the observation of  $\omega$  (e.g. customers being serviced outside their time windows). In particular, one of the recourse actions consists of servicing a customer on a different location when the location given by the first-stage is not feasible given the revealed travel times.

In the following section, we describe each considered recourse action.

### 3.4 Recourse actions

Each non-serviced customer (location) potentially leads to an extra cost ( $\Lambda_c$  in the two-stage formulation) representing, for example, the cost of outsourcing the service. Moreover, routes exceeding the time horizon  $T$  incur an overtime penalty proportionally to the extra time. For all proposed actions, a vehicle initially follows its a priori route visiting customer locations within their time windows (in case of early arrival, waiting is allowed). First, we describe three recourse actions not utilizing opportunities to service a customer at a different location than that of the first-stage route.

- *Do nothing* ( $R_0$ ): No corrective action is taken i.e., the second-stage route remains the same as in the first-stage. The penalty  $\Lambda_c$  is incurred for all customers  $c \in C$  serviced outside their time windows.
- *Skip next customer* ( $R_1$ ): Given an a priori route, the realized travel times could lead to infeasibilities serving one or more customers within their time windows. If serving the next customer in the route is not possible (within the time window of the selected location), that customer is skipped and a penalty is incurred. Customers in the route continue to be skipped until a customer location able to be visited within its corresponding time windows is found (given the a priori route). If no customer location is found, the vehicle returns to the depot. Figure 2a illustrates an example for a route visiting four customers. After servicing customer  $c_2$ , visiting  $c_3$  within the time windows of the selected location is not possible and the customer is skipped. Similarly, it is not possible to visit

$c_4$  (from  $c_2$ 's location) within the corresponding time window of the selected location and customer  $c_4$  is also skipped. Visiting customer  $c_5$  at the selected location is feasible and, after servicing  $c_2$ , the vehicle visits customer  $c_5$ .

- *Skip customers* ( $R_2$ ): Similar to the previous recourse action, customers are skipped whenever the first-stage route leads to a late visit, but the decision on which customers to skip is optimized. Given a first stage route  $r = (v_0 = 0, v_1, v_2, \dots, v_{|r|} = n + 1)$ , where  $v_i \in N \cup \{0, n + 1\}$ ,  $i = 0, \dots, |r|$  are the locations visited by  $r$ , and the realized travel times  $\tilde{\omega}$ , define the following dynamic program:

$$z(v_i, v_j, \tau_i) = \begin{cases} \beta \max\{\tau_i + \tilde{\omega}_{v_i, v_j} - T, 0\} & \text{if } v_j = n + 1 \\ \min\{z(v_j, v_{j+1}, \max\{\tau_{\pi(v_i)} + \tilde{\omega}_{v_i, v_j}, \tilde{a}_{v_j}\}), & \text{if } \tau_{\pi(v_i)} + \tilde{\omega}_{v_i, v_j} \leq \tilde{b}_{v_j} \\ z(v_i, v_{j+1}, \tau_{\pi(v_i)}) + \Lambda_{\pi(v_j)}\} & \\ z(v_i, v_{j+1}, \tau_{\pi(v_i)}) + \Lambda_{\pi(v_j)} & \text{otherwise} \end{cases} \quad (26)$$

where  $z(v_i, v_j, \tau_i)$  is the minimum total penalty cost incurred for skipping customers in the (partial) route  $(v_i, \dots, v_j)$ , where the earliest time at  $v_i$  is  $\tau_i$ . For route  $r$ , the minimum value is obtained by solving  $z(0, v_1, a_0)$  and the corresponding set of skipped customers can be obtained by backtracking the optimal path in the recursion.

Since customers in the VRPRDL-STT have different possible locations for delivery (with non-overlapping time-windows), a natural recourse action consists in attempting delivery at another location in the customer's itinerary.

- *Reschedule next customer* ( $R_3$ ): Similar to the *skip next customer* recourse, customers are visited following the locations given by the first stage route. If servicing the next location in the a priori route, within the location's time window, is not possible, instead of skipping this customer, a delivery to a different location is evaluated. If service is possible within the time window of another location in the customer's itinerary, then this location is selected and the customer is visited there. In case multiple locations can be selected, the one with the minimum detour time is chosen. If no location can be selected, the customer is skipped. Figure 2b depicts the same a priori route as in Figure 2a but applies this recourse in the second stage. Customer  $c_3$  is skipped, as no other location in the set  $N_{c_3}$  can be visited within the corresponding time windows. Customer  $c_4$  would be skipped if only the location selected in the first stage route could be visited. However, by selecting a different location in the set  $N_{c_4}$  (in the example, visiting  $i_3^{c_4}$  instead of  $i_1^{c_4}$ ), the vehicle can still deliver to customer  $c_4$ .
- *Reschedule customers* ( $R_4$ ): Similarly to the previous recourse action, a different location other than that in the first-stage solution is selected for customers that would otherwise not be serviced by following the first-stage route. However, the set of customers to skip, and evaluate a new location for servicing, is obtained by solving the dynamic program defined in Recourse 2. All locations corresponding to skipped customers are removed from the route before trying to reinsert those customers again, but at a different location, in the route. Given a skipped customer,  $c \in C$ , a location  $i \in N_c$  is inserted at the best position in the route, minimizing the required detour, and the best feasible insertion is

executed. If no location  $i \in N_c$  is feasible to be inserted, customer  $c$  is skipped in the route.

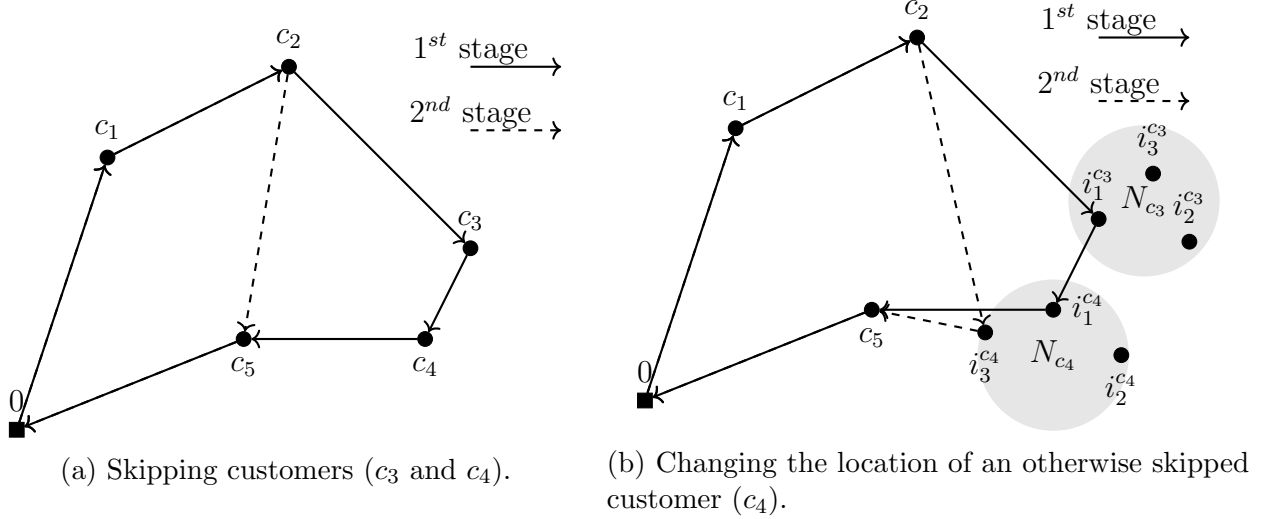


Figure 2: Recourse actions applied to a first-stage route visiting five customers. In 2a, customers are skipped if delivery occurs outside the location time windows ( $R_1$ ) and, in 2b, a different location might be selected for an otherwise skipped customer ( $R_3$ ).

## 4 Sample Average Approximation

The Sample Average Approximation is a framework to solve stochastic (discrete) optimization problems by Monte-Carlo simulation proposed by Kleywegt et al. (2002). A computational study using the method for solving two-stage stochastic routing problems in the form  $\min_{\mathbf{x} \in X} \mathbf{w}'\mathbf{x} + \mathbb{E}[R(\mathbf{x}, \boldsymbol{\omega})]$  was conducted by Verweij et al. (2003). The idea is to approximate the expected value function  $\mathbb{E}[R(\mathbf{x}, \boldsymbol{\omega})]$  by the corresponding sample average function  $z(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^N R(\mathbf{x}, \tilde{\boldsymbol{\omega}}^i)$ , where  $\Omega = \{\tilde{\boldsymbol{\omega}}^1, \dots, \tilde{\boldsymbol{\omega}}^N\}$  is a set of realizations (scenarios) of the random vector  $\boldsymbol{\omega}$ , and solve a Sample Average Approximation (SAA replication) problem  $\min_{\mathbf{x} \in X} v_\Omega(\mathbf{x})$ ,  $v_\Omega(\mathbf{x}) = \mathbf{w}'\mathbf{x} + z(\mathbf{x})$ , considering not the full support of  $\boldsymbol{\omega}$  – which can grow exponentially with the dimension of  $\mathbf{x}$  – but a smaller set of realizations. A large set of realizations,  $\Omega' = \{\tilde{\boldsymbol{\omega}}^{1'}, \dots, \tilde{\boldsymbol{\omega}}^{N'}\}$ ,  $N' \gg N$ , is then used to approximate the true expected value (solution gap) of the solution obtained by solving the SAA problem. The procedure is repeated until a certain criteria is met (e.g. the maximum number of replications, the optimality gap is small enough).

Algorithm 1 illustrates the main steps of our SAA implementation. Usually, implementations of the SAA framework proposed by Kleywegt et al. (2002) solve a fixed number,  $M$ , of SAA replications, generally using a fixed sample size  $|\Omega| = N$ . In contrast, in our implementation, the number of replications solved is not fixed and the sizes of the samples used to solve each replication is adjusted accordingly to the performance of the method for a current value, namely, the gap estimator,  $\epsilon_m$ , after solving the  $m^{th}$  SAA replication. Kleywegt et al. (2002) utilize  $\epsilon = v_{\Omega'}(\mathbf{x}) - \hat{v}_\Omega$  as an estimator of the (true) optimality gap  $v_{\Omega'}(\mathbf{x}) - v^*$ , where  $v^*$  is the optimal cost of the problem considering all possible realizations of  $\boldsymbol{\omega}$ . By solving each

---

**Algorithm 1:** Overview of the (proposed) SAA framework implementation

---

**Input** : Initial sample size  $N$ ;  $N'$ , the large set of realizations  $\Omega' = \{\tilde{\omega}^1, \dots, \tilde{\omega}^{N'}\}$ ;  
Number of replications  $l$  to check for convergence

**Output:** Solution  $\mathbf{x}^*$

```

1   $m \leftarrow 1$ 
2  Generate  $\Omega^m = \{\tilde{\omega}^1, \dots, \tilde{\omega}^N\}$ 
3  Solve the SAA problem over  $\Omega^m$ , with objective  $v_{\Omega^m}$  and solution  $\mathbf{x}^m$ 
4  Evaluate  $\mathbf{x}^m$  on  $\Omega'$ :
5     $v_{\Omega'}(\mathbf{x}^m) \leftarrow \mathbf{w}'\mathbf{x}^m + \frac{1}{N'} \sum_{i=1}^{N'} R(\mathbf{x}^m, \tilde{\omega}^{i'})$ 
6  Compute the average of solutions found in previous iterations
7     $\hat{v}_{\Omega} \leftarrow \frac{1}{m} \sum_{i=1}^m v_{\Omega^i}$ 
8  Compute the SAA gap estimate:
9     $f^* \leftarrow \min_{i=1, \dots, m} v_{\Omega'}(\mathbf{x}^i)$ ;
10    $\epsilon_m \leftarrow \frac{f^* - \hat{v}_{\Omega}}{f^*}$ 
11 Compute variance of  $\hat{v}$  over the past  $l$  replications
12    $\sigma_{\hat{v}_{\Omega}}^2 = \frac{1}{(L-1)} \sum_{i=m-l}^m (\hat{v}_{\Omega} - v_{\Omega^i})^2$ 
13    $\sigma_{\epsilon_m}^2 = \frac{1}{(L-1)} \sum_{i=m-l}^m (\hat{\epsilon} - \epsilon_i)^2$ 
14 Check convergence and update the sample size:
15 if  $\sigma_{\hat{v}_{\Omega}} \leq 0.01$  then
16   if  $\epsilon_m \leq 0.05$  then
17     Generate a new (independent) sample set  $\Omega'$ 
18     return  $\mathbf{x}^* = \min_{i=1, \dots, m} v_{\Omega'}(\mathbf{x}^i)$ 
19   else
20      $N \leftarrow N + \Delta$ 
21  $m \leftarrow m + 1$ 
22 goto 2

```

---

SAA replication to optimality, it can be shown that  $v^* - \mathbb{E}[\hat{v}_{\Omega}]$  is monotonically decreasing in  $N$  i.e.,  $\hat{v}_{\Omega}$  is a statistical lower bound for  $v^*$  (Verweij et al., 2003). Since in our approach individual SAA problems are solved heuristically,  $\hat{v}_{\Omega}$  is not necessarily a valid lower bound and, moreover, the lower bound estimator  $\hat{v}_{\Omega}$  tends to overestimate the true lower bound. However, we still compute and use the estimator  $\hat{v}_{\Omega}$  in order to evaluate the performance of the SAA problems given the current sample size,  $N$ , and to adjust (increase) the sample size throughout the method, but only after a certain level of convergence (variance of the values observed in the past  $l$  replications) is achieved.

Choosing the sizes  $N$  and  $N'$  is a trade-off between solution quality and computational efficiency in solving the SAA problems. Observe that solving each SAA problem becomes more time consuming as  $N$  increases, but the estimated lower bound,  $\hat{v}_{\Omega}$ , tends to be stronger and, consequently, the SAA gap tends to be smaller. In our implementation, we use a fixed  $|\Omega'| = N'$  throughout the algorithm, and start solving the SAA problems over small sample sets  $|\Omega| = N$ . An integer parameter  $\Delta$  is used to control the extent by which to increase  $N$  (how many additional scenarios to consider). When the solutions obtained by replications with the same sample size  $N$  converge, the gap estimator  $\epsilon_m$  is evaluated and, in case it is not yet below the

tolerance,  $N$  is increased by  $\Delta$  and new replications are solved, until convergence. Otherwise, all solutions obtained by solving each SAA replication are evaluated over a new (independent) set  $\Omega'$  and the best solution is returned.

In Section 5, we conduct a series of experiments to assess the implementation decisions taken in our proposed SAA.

## 4.1 Solving the SAA Problem

To solve the SAA problem  $\min_{\mathbf{x} \in X} \mathbf{w}'\mathbf{x} + \frac{1}{N} \sum_{i=1}^N R(\mathbf{x}, \tilde{\omega}^i)$ , we resort to an adaptation of the heuristics proposed by Reyes et al. (2017) which considers travel time uncertainty in the arcs. We did so by adding the sample average function to the objective and, as a result, we also modified the local search operators such that the marginal cost of an insertion considers both the changes in distance and recourse costs over  $N$  scenarios  $\Omega = \{\tilde{\omega}^1, \dots, \tilde{\omega}^N\}$ . In particular, computing the change in the recourse costs before and after an operator is applied requires the evaluation of the modified solution (route) on the sample set  $\Omega$ . Algorithm 2 gives an overview of how second-stage recourse costs are computed for a given first-stage solution and recourse policy,  $R$ , considering a sample of  $K$  travel time realizations.

---

**Algorithm 2:** Evaluating second stage costs for a first-stage solution  $\mathbf{x}$ .

---

**Input** : First stage routing solution  $\mathbf{x}$ ; a sample  $\Omega = \{\tilde{\omega}^1, \dots, \tilde{\omega}^K\}$  of travel time realizations; a recourse action  $R \in \{R_0, R_1, R_2, R_3, R_4\}$ .

**Output:** The average recourse cost  $\frac{1}{N} \sum_{i=1}^K R(\mathbf{x}, \tilde{\omega}^i)$

```

1  $R(\mathbf{x}, \Omega) \leftarrow 0$ 
2 for each scenario  $\tilde{\omega}^i \in \Omega$  do
3   for each route,  $r$ , in  $\mathbf{x}$  do
4     Apply recourse  $R$  to  $r$ , considering travel times  $\tilde{\omega}^i$ , to obtain a new route  $r'$ 
5      $U \leftarrow$  set of customers visited by  $r$  but not visited following  $r'$ 
6     for each customer  $c \in U$  do
7        $R(\mathbf{x}, \Omega) \leftarrow R(\mathbf{x}, \Omega) + \Lambda_c$ 
8        $\lambda_{n+1} \leftarrow$  arrival time at the target depot for route  $r'$ 
9        $R(\mathbf{x}, \Omega) \leftarrow R(\mathbf{x}, \Omega) + \beta \max\{0, \lambda_{n+1} - T\}$ 
10 return  $\frac{R(\mathbf{x}, \Omega)}{K}$ 
```

---

Given a realization of travel times, one of the recourse decisions described in Section 3.4 is applied to each a priori route in the first-stage solution  $\mathbf{x}$  (Line 4), and a new route (second-stage route) is obtained in which customers are skipped and/or a different customer location is selected for service. The penalty  $\Lambda_c$  is incurred for every customer  $c \in U \subseteq C$  not visited in the second-stage route (Line 7) as well as the overtime cost (Line 9). Finally, Algorithm 2 returns the average recourse cost for solution  $\mathbf{x}$  computed over all travel time scenarios in  $\Omega$  (Line 10).

## 5 Computational Evaluation

In this section, we present a set of computational experiments conducted to evaluate the performance of the proposed framework in solving the VRPRD-STT and to assess the benefits of

trunk delivery considering unknown travel times at the time of planning. Our algorithms are coded in Java and all experiments are executed on an Intel Xeon E5-2666 v3 CPU @3.5GHz machine, 15GiB, running Ubuntu Server 18.04.

In our experiments, unless explicitly stated otherwise, we use the following input parameters. The number of samples used to solve each SAA replication is initially set to  $N = 1$  and the increase parameter  $\Delta$  is set to  $5\lceil\frac{\epsilon_m}{0.05}\rceil$ , e.g., the higher the relative difference between current and desired (estimate) gaps the larger the increase in the number of samples. The set  $\Omega'$  has size  $|\Omega'| = 10000$ . Convergence of the estimators (gap and lower bound) are checked considering the variance over the last  $l = 15$  replications and the algorithm terminates when convergence is attained and the gap estimator is below 0.05. We consider a congestion level of  $\eta = 0.35$  (representative for the average congestion level of the top 100 congested cities worldwide (Tom Tom, 2016)). The transportation cost matrix  $\mathbf{w} = (w_{ij})_{(i,j) \in A_R}$  is defined by the travel distance between any two locations  $i$  and  $j$ . The overtime penalty is  $\beta = 2$  and the maximum allowed overtime is set to 120 time units (two hours). We define the penalty cost of not servicing customer  $c \in C$ , in the second stage, as  $\Lambda_c = 10F_c$ , where  $F_c$  is the total (distance) cost of a single route from the depot to the home location of customer  $c$  and back to the depot e.g., the cost of a subcontracted, dedicated service. For a solution  $\mathbf{x}$ , we define  $\mathbf{w}'\mathbf{x}$  and  $\mathbb{E}[R, \mathbf{x}]$  as the corresponding first and second-stage costs, respectively.

## 5.1 Instance Description

To evaluate the impact of stochastic travel times in solving the VRPRDL, we perform computational experiments using problem instances introduced by Reyes et al. (2017). In particular, we consider their set of *general* instances, with 15, 20 and 30 customers, each with up to five roaming delivery locations, a time horizon of  $T = 720$  (12 hours), a single depot (locations 0 and  $n + 1$  are the same) at the center of the region under consideration, and vehicles with capacity  $Q = 750$ . The geographic profile of each customer,  $c \in C$ , is as follows: the roaming delivery locations of  $c$ ,  $N_c$ , are centered around the customer's home location and all are reachable, within the time horizon, from the depot; the itinerary of  $c$  starts at its home location, visits each roaming location and ends at home. Time windows (i.e., the time customer  $c$  spends at each location, being available for delivery) are generated by subtracting the total time spent by travelling from the time horizon and allocating the remaining time, in uniformly random lengths, to each location  $i \in N_c$ . Observe that if  $|N_c| = 1$  then customer  $c$  is available during the whole time horizon exclusively at home.

Table 2 reports the characteristics of each instance considered in the computational experiments. Each row represents an instance and depicts the number of customers ( $|C|$ ), the total number of locations ( $\sum_c N_c$ ) and the average time, as a percentage of the time horizon  $T$ , that customers are available for delivery (Av.), where customer time windows are defined by deterministic (expected) travel times. We also distinguish between customers available only at home ( $|N_c| = 1$ , available over the full time horizon) and customers available at roaming delivery locations ( $|N_c| > 1$ ). Thus, in the column Roaming, we report the average (Avg.), the minimum (Min) and maximum (Max) percentage of time that customers with roaming locations are available for delivery.

A disruption on an arc might alter the customer availability at the locations for trunk delivery (see Figure 1). In an extreme case, this means that delivery at some locations becomes infeasible. To give an insight on how changes in travel times might affect the availability (time



windows at each location in the itinerary) we also report, in column Stochastic, the number of locations ( $Avg_m$ ) that are unavailable once travel times are revealed and the time available for delivery (Av), as the average values over 10000 different travel time realizations.

Instance	All customers (deterministic)			Roaming customers (deterministic)			Stochastic	
	$ C $	$\sum_c N_c$	Av.(%)	Avg(%)	Min (%)	Max (%)	$Avg_m$	Av(%)
$I_0$	15	63	63.1	47.7	4.9	81.4	5.4	60.9
$I_1$	15	58	69.1	52.2	31.0	90.4	3.9	66.2
$I_2$	15	53	71.3	53.5	7.9	98.6	7.4	68.1
$I_3$	15	51	64.2	39.1	8.3	82.4	6.1	61.6
$I_4$	15	53	69.9	55.7	14.2	96.9	3.6	69.8
$I_5$	20	67	69.9	49.1	16.3	99.7	5.2	68.1
$I_6$	20	69	67.7	40.7	7.5	97.2	9.3	65.8
$I_7$	20	81	60.0	41.3	4.4	64.2	10.3	57.3
$I_8$	20	77	53.9	36.6	1.0	85.0	10.3	51.6
$I_9$	20	64	69.5	52.1	14.2	98.6	6.9	67.3
$I_{10}$	30	104	64.2	50.2	3.1	98.1	6.7	61.8
$I_{11}$	30	114	56.4	41.8	0.6	89.4	14.9	53.6
$I_{12}$	30	119	57.0	44.9	3.8	83.3	13.6	54.0
$I_{13}$	30	108	63.6	47.1	12.4	91.7	9.4	61.7
$I_{14}$	30	125	56.7	46.8	4.0	96.3	13.7	54.1
$I_{15}$	30	120	61.5	48.7	2.8	93.8	6.3	58.8
$I_{16}$	30	131	53.0	46.3	7.5	93.5	13.3	50.2
$I_{17}$	30	107	61.1	43.4	0.6	64.4	9.9	58.7
$I_{18}$	30	99	65.0	49.1	7.6	85.3	10.3	62.5
$I_{19}$	30	76	78.1	50.0	4.9	91.4	7.9	76.8

Table 2: Description of a subset of the general VRPRDL instances proposed by Reyes et al. (2017).

## 5.2 Generating Travel Times Scenarios

For this work, we follow an approach similar to Taş et al. (2013), Jabali et al. (2015) and Vareias et al. (2017). We model the (stochastic) time to traverse an arc  $(i, j) \in A$  as a random variable given by  $t_{ij} + \delta_{ij}$ , where  $\delta_{ij} \geq 0$  is the duration of the stochastic disruption on arc  $(i, j)$  and  $t_{ij}$  is a deterministic travel time, used in the first stage. Specifically, we consider that  $\delta_{ij}$  follows a gamma distribution with a given shape parameter,  $k$ , and scale,  $\theta_{ij}$ , depending on the deterministic travel time value of the arc. Gamma distributions are used to describe stochastic travel times in the literature, as it follows convolution and non-negativity properties. The parameters  $k$  and  $\theta_{ij}$  allow for the generation of scenarios considering the degree by which travel times vary, adjusted by the *coefficient of variation* ( $\hat{c}_v$ ).

Let  $\eta \geq 0$  be a congestion level, representing the (expected) increase in travel time proportional to the travel time used in the first stage i.e., the travel time of arc  $(i, j)$  after a disruption incurs an expected increase of  $\mathbb{E}[\delta_{ij}] = \eta t_{ij}$ . Thus, if  $\delta_{ij} \sim G(k, \theta_{ij})$ , we have:

$$\mathbb{E}[\delta_{ij}] = k\theta_{ij} = \eta t_{ij} \quad (27)$$

$$Var(\delta_{ij}) = k\theta_{ij}^2 \quad (28)$$

We derive parameters  $k$  and  $\theta_{ij}$  for a given value  $\hat{c}_v$  as follows:

$$\hat{c}_v = \frac{\sqrt{k\theta_{ij}^2}}{k\theta_{ij}} \implies k = \frac{1}{\hat{c}_v^2}, \theta_{ij} = \eta t_{ij} \hat{c}_v^2 \quad (29)$$

519 A scenario  $\tilde{\omega}$  is, then, an assignment of a random value to each arc  $(i, j) \in A$ , with  $\tilde{\omega}_{ij} =$   
 520  $t_{ij} + \delta_{ij}$ , where  $\delta_{ij}$  is drawn from the gamma distribution  $G(k, \theta_{ij})$ , given a value of  $\hat{c}_v$ . Figure 3  
 521 illustrates the probability density of the gamma distribution used to model the disruption on a  
 522 particular arc  $(i, j) \in A$  with  $t_{ij} = 60$  and  $\eta = 0.35$ , considering different (squared) coefficient  
 523 of variation values:  $\hat{c}_v^2 = 0.0625$  (Figure 3a),  $\hat{c}_v^2 = 0.25$  (Figure 3b) and  $\hat{c}_v^2 = 1.00$  (Figure 3c).  
 524 Within the SAA framework, the objective value (12) of the solution  $\mathbf{x}^i$  obtained after solving  
 525 the  $i^{th}$  SAA replication, is approximated by  $v_{\Omega'}(\mathbf{x}^i)$ , taking a very large sample  $\Omega'$ . Thus, Figure  
 526 3 presents an interval frequency for  $|\Omega'| = 10000$  values sampled from the distribution.

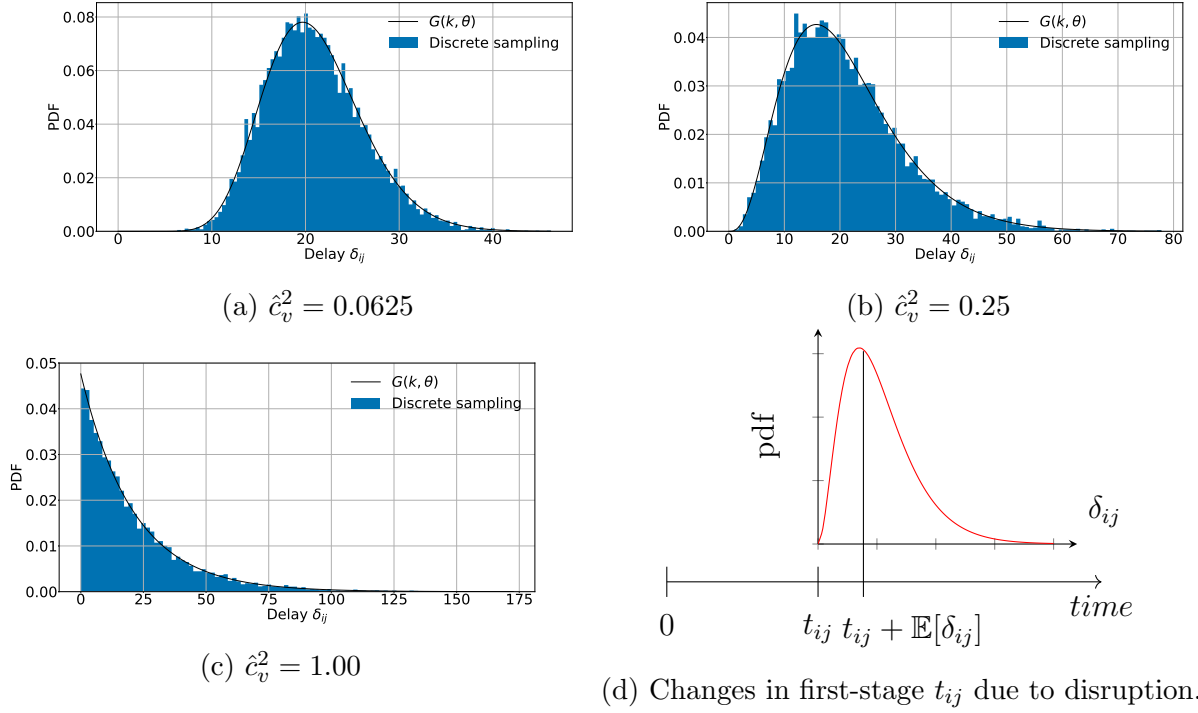


Figure 3: Disruption,  $\delta_{ij}$ , on arc  $(i, j)$ ,  $t_{ij} = 60$  and  $\eta = 0.35$ , sampled with different  $\hat{c}_v^2$  values.

527 In this specific example, the disruption  $\delta_{ij}$  has the same expected value,  $\mathbb{E}[\delta_{ij}] = 21$ , regard-  
 528 less of the coefficient value used to obtain the parameters  $k$  and  $\theta_{ij}$  of the gamma distribution.  
 529 However, the samples derived from each distribution differ significantly. In particular, observe  
 530 that the maximum disruption observed for  $\hat{c}_v^2 = 1.00$  is approximately four (resp. two) times  
 531 the maximum disruption observed for  $\hat{c}_v^2 = 0.0625$  (resp.  $\hat{c}_v^2 = 0.25$ ). The higher  $\hat{c}_v^2$ , the higher  
 532 the number of samples with low probability, but with larger disruption values. Scenarios drawn  
 533 from a distribution with  $\hat{c}_v^2 = 0.0625$  represent short length disruptions occurring frequently  
 534 (e.g., sudden increase in traffic as a consequence of events on adjacent streets), whereas val-  
 535 ues sampled from a distribution with  $\hat{c}_v^2 = 1.00$  reflect severe disruptions with low probability  
 536 of happening (e.g., accidents blocking one or more lanes, severe speed reduction due to road  
 537 condition).

We consider that the travel times of arcs in the instances introduced by Reyes et al. (2017) already account for expected disruptions i.e., the deterministic travel time,  $\bar{t}_{ij}$  of arc  $(i, j) \in A$  is  $\bar{t}_{ij} = t_{ij} + \mathbb{E}[\delta_{ij}]$ . When solving an instance taking into account the stochastic disruptions, the travel time of an arc  $(i, j) \in A$  in the first-stage is  $t_{ij} = \frac{\bar{t}_{ij}}{(1+\eta)}$ , and the stochastic disruption,  $\delta_{ij}$ , in the second-stage is sampled from a gamma distribution, as shown previously, i.e.,  $t_{ij}$  is the travel time to traverse arc  $(i, j) \in A$  without any disruption, and  $\mathbb{E}[\delta_{ij}] = \eta t_{ij}$ . Figure 3d illustrates the travel and disruption times for an arc  $(i, j) \in A$ . In the remaining sections, unless specified otherwise, we use  $\hat{c}_v^2 = 0.25$ .

### 5.3 The Value of a Stochastic Solution

To assess the value of stochastic information, we obtain a stochastic solution,  $\mathbf{x}^s$ , using our SAA framework, as well as a deterministic solution,  $\mathbf{x}^d$ , in which all random variables are replaced by their *expected values*. Next, we compare the routing costs (first stage), as well as the recourse costs (second stage) for both solutions. The expected recourse cost for a solution is obtained through Algorithm 2, using a large scenario sample size  $|\Omega| = 10000$ . Through this experiment, we show the extent to which first-stage decisions and costs of a stochastic solution change as a consequence of incorporating uncertainty in the travel times. To facilitate a fair comparison, we require that the number of routes in the stochastic solution does not exceed the number of routes used in the deterministic solution. As elaborated in Section 4, we use our adapted implementation of the local search heuristic proposed by Reyes et al. (2017) to compute both  $\mathbf{x}^d$  and  $\mathbf{x}^s$ . To calculate  $\mathbf{x}^s$ , we use  $\mathbf{x}^d$  as a starting solution to our heuristic.

Figure 4 depicts the comparison of  $\mathbf{x}^d$  and  $\mathbf{x}^s$  for the instances in Table 2 using recourse  $R_0$ . Recall that in  $R_0$  no corrective actions are taken: the second-stage costs solely include penalties incurred for missed customers and overtime. In this experiment, the scenarios are generated using a coefficient of variance  $\hat{c}_v^2 = 0.25$ . The results for  $\mathbf{x}^s$  are averaged over 3 runs of the SAA framework.

For each instance, Figure 4 shows the routing costs  $\mathbf{w}'\mathbf{x}$ , and the expected recourse costs,  $\mathbb{E}[R_0, \mathbf{x}]$ , for both the deterministic solution  $\mathbf{x}^d$  and the stochastic solution  $\mathbf{x}^s$ . The total cost of a solution is given by  $\mathbf{w}'\mathbf{x} + \mathbb{E}[R_0, \mathbf{x}]$ . Figure 4 also shows the value of the stochastic solution computed as  $vss = 100 \times \frac{(v_{\Omega'}(\mathbf{x}^d) - v_{\Omega'}(\mathbf{x}^s))}{v_{\Omega'}(\mathbf{x}^d)}$ , where  $v_{\Omega'}(\mathbf{x}^d)$  and  $v_{\Omega'}(\mathbf{x}^s)$  are the total costs of the deterministic and stochastic solutions, respectively.

As can be observed in Figure 4, the total costs of the deterministic solutions are significantly higher than the costs of the stochastic solutions. In fact, using our stochastic framework, we realize a cost reduction of nearly 42% compared to deterministic solutions. The difference in first-stage routing costs is small (on average 1%) but the routing plans differ significantly. The main changes stem from vehicle-customer assignment decisions, the order of service and the selected service locations.

In Figure 5 we repeat the same experiment for recourse actions  $R_1$ - $R_4$ . Again, we observe that the stochastic solutions drastically improve over the deterministic solutions: the average cost reductions are 38%( $R_1$ ), 33%( $R_2$ ), 40% ( $R_3$ ), and 32%( $R_4$ ). Note that in these experiments, the deterministic solution,  $\mathbf{x}^d$ , and its routing cost,  $\mathbf{w}'\mathbf{x}^d$ , remain constant independent of the recourse action being used; only the expected recourse cost  $\mathbb{E}[R_i, \mathbf{x}^d]$  changes due to its dependency on  $R_i$ . When comparing the expected recourse costs  $\mathbb{E}[R_i, \mathbf{x}^d]$  for  $i = 1, \dots, 4$  and fixed  $\mathbf{x}^d$ , we observe that on average the recourse costs of  $R_4$  are lowest, followed by  $R_3$ ,  $R_2$  and finally  $R_1$ .

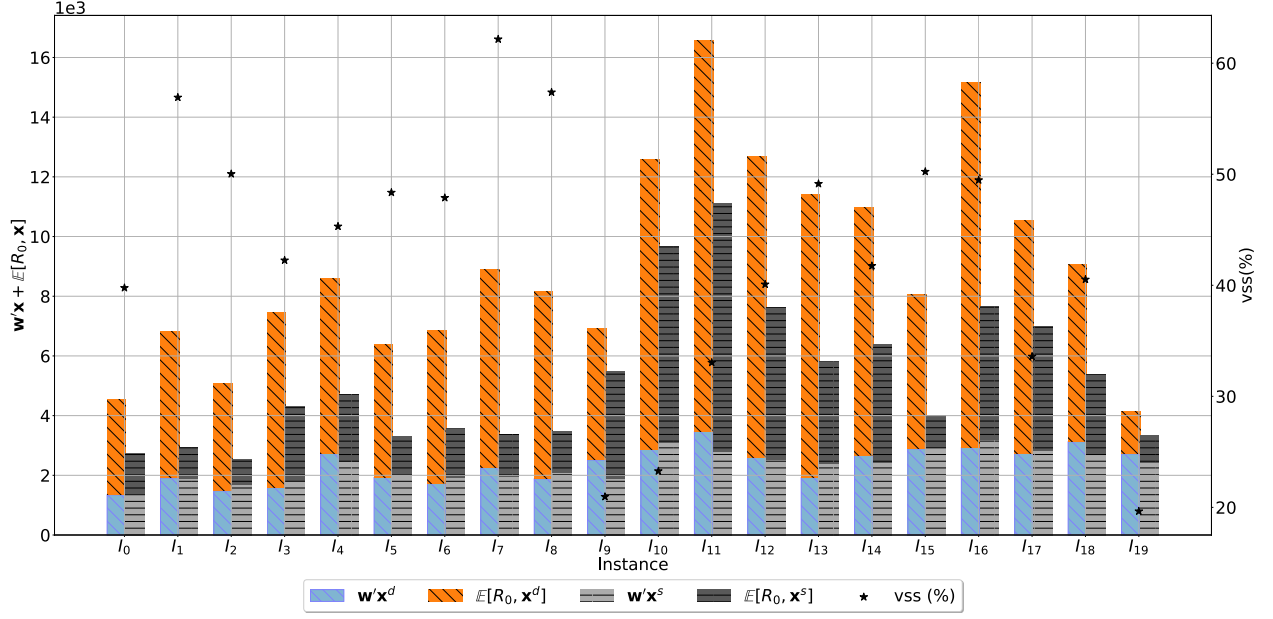


Figure 4: The value of incorporating stochastic information. Second-stage costs are derived from recourse  $R_0$ , travel times scenarios are derived from distributions having coefficient of variation  $\hat{c}_v^2 = 0.25$ .

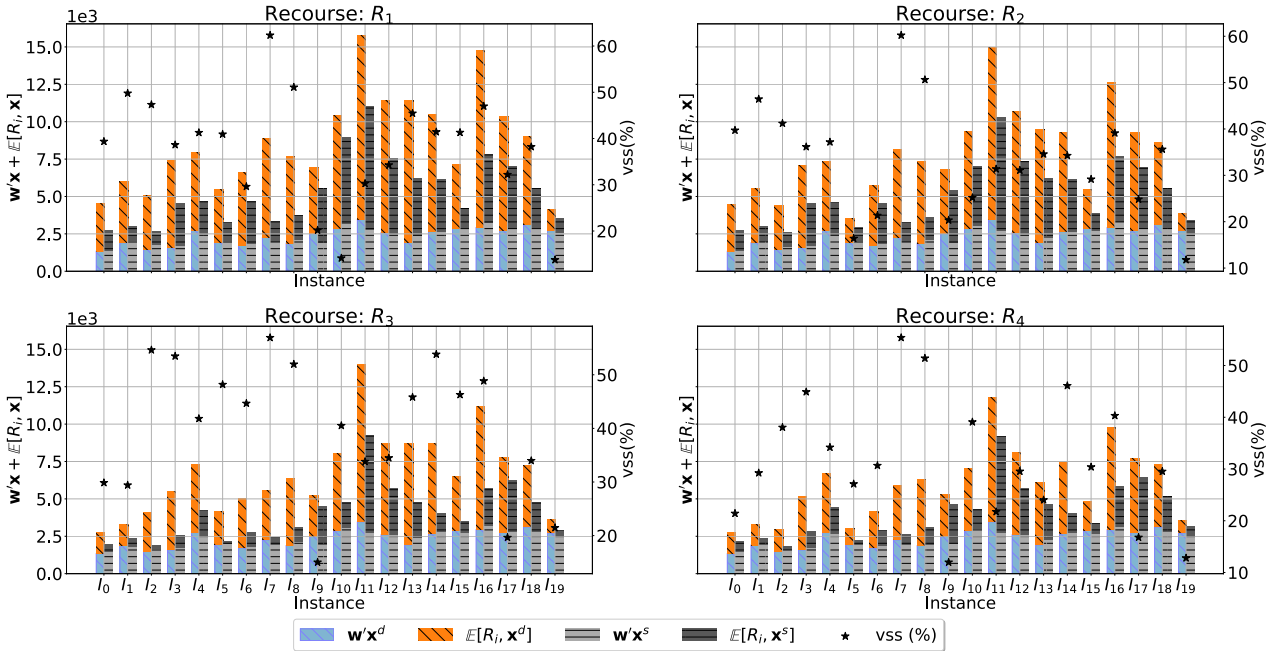


Figure 5: The value of incorporating stochastic information to solve the VRPRDL using recourse actions  $R_1, R_2, R_3$  and  $R_4$ . Travel times scenarios derived from distributions having coefficient of variation  $\hat{c}_v^2 = 0.25$ .

## 5.4 Recourse Policy Comparison

A mutual comparison of recourse policies  $R_1$  and  $R_2$ , as well as  $R_3$  and  $R_4$  is provided in Figure 6. Per instance, the average results over three runs of the SAA algorithm are shown.

Recall from Section 3.4 that recourse policies  $R_1$  and  $R_3$  *greedily* skip or reschedule customers that cannot be reached within their time windows, whereas policies  $R_2$  and  $R_4$  use a more involved DP to compute the best subset of customers in a given route to skip or reschedule. From Figure 6, we confirm that recourse policy  $R_2$  outperforms its simpler counterpart ( $R_1$ ): the costs of  $R_2$  are on average 4% lower than  $R_1$ . The difference in costs between policies  $R_3$  and  $R_4$  are negligible. These cost reductions come with an increase of computation time of 50–60%. The limited performance difference between  $R_2$  and  $R_1$  is explained by the fact that the average number of customers per route is quite low (4–5 customers). As a result, determining which customer(s) to skip, either greedily or via a DP, does not lead to drastically different routes. Therefore, larger benefits can be obtained with recourse  $R_2$  when the number of customers per route increase, but this comes with a significant increase in computation times. Specifically, comparing the results on the instance for which recourse  $R_2$  improves over  $R_1$  the most (20%) and the results on the instance with the least improvement ( $< 1\%$ ),  $R_2$  outperforms  $R_1$  when the solution contains slightly longer routes (5–6 customers), whereas the benefit is minimal when there are mostly short routes (e.g., most routes services less than 4 customers).

Figure 7 compares policies  $R_1$  and  $R_3$ , investigating the benefit of being able to visit a customer at a different location than the one selected in the a priori plan. Policy  $R_3$ , in contrast to  $R_1$ , attempts to reschedule a customer at an alternative location in case of a missed delivery. Solution costs obtained with policy  $R_3$  are on average 25% lower than the costs obtained with policy  $R_1$ : both the first stage costs and the second stage costs are lower (resp. 1% and 44%). Do note that evaluating recourse  $R_3$  is generally more complex than evaluating  $R_1$ , and, as a result, evaluating the second-stage costs using  $R_3$  takes about twice the execution time required to evaluate recourse  $R_1$ . In practice, this computation time increases proportional with the number of alternative locations in the itinerary of a customer.

Based on these results, we conclude that recourse policy  $R_4$  incurs in the highest computational costs. A significant reduction in computation time is realized with recourse policy  $R_3$ , with similar solution quality. If even faster computation times are required, then recourse action  $R_1$  is recommended, which still achieves a 37% improvement over a deterministic model which only takes expected values into account.

## 5.5 The Impact of Travel Time Variance

To assess the impact of travel time variance, we conduct an experiment in which the stochastic travel times are drawn from different gamma distributions: one with a *low* variance ( $\hat{c}_v^2 = 0.0625$ ) and another one with a *high* variance ( $\hat{c}_v^2 = 1.00$ ). The results averaged over *all* instances are reported in Table 3. Again the SAA algorithm has been invoked 3 times per instance.

As can be observed from Table 3, the solution costs  $v_{\Omega'}(\mathbf{x}^*)$  increase significantly when the variance increases:  $v_{\Omega'}(\mathbf{x}^*)$ , for the high variance instances, is almost twice the value of the low variance instances. On average, the second-stage recourse costs account for 31% (low variance) resp. 63% (high variance) of these total costs. Next to an increase in costs, we witness a decrease in the Value of the Stochastic Solutions (VSS), from 38.4% for the low variance instances to 33.7% for the high variance instances. Finally, we observe a significant

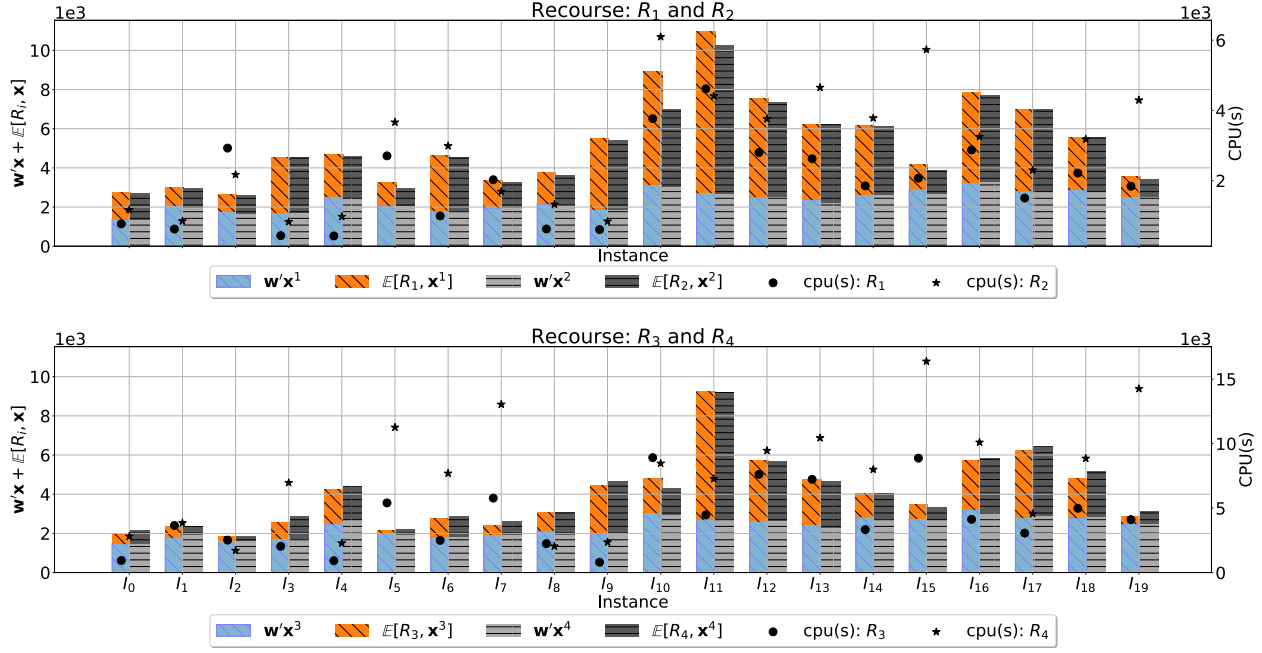


Figure 6: Comparison of solutions obtained by recourse actions  $R_1$  and  $R_3$  compared to solving the dynamic programming ( $R_2$  and  $R_4$ ) model (26).

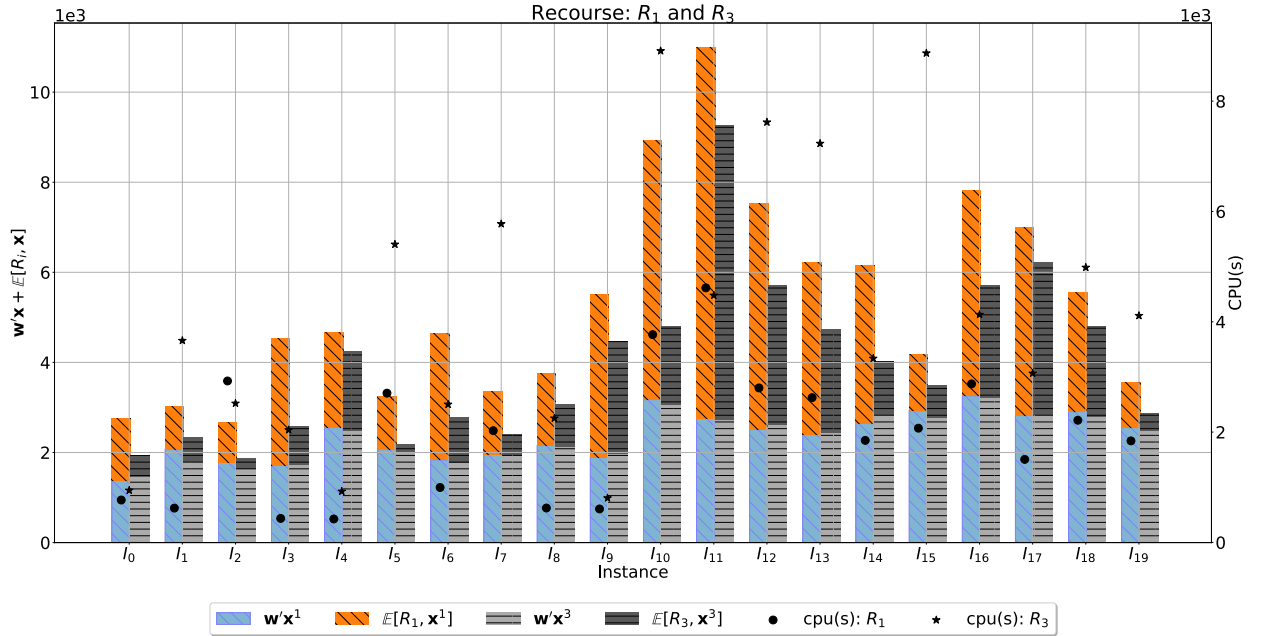


Figure 7: Savings by changing the service location of an otherwise skipped customer.

	$\hat{c}_v^2 = 0.0625$			$\hat{c}_v^2 = 1.00$		
	$v_{\Omega'}(\mathbf{x}^*)$	T(s)	$vss(\%)$	$v_{\Omega'}(\mathbf{x}^*)$	T(s)	$vss(\%)$
Average	3,107	1,920	38.4	5,546	7,881	32.3

Table 3: Impact of travel time scenarios generated with different coefficient of variance.  $R_3$  is applied as second-stage recourse action.

increase in computation times when the variance increases. This increase is attributed to the fact that in case of a high variance, more SAA replications are needed with a larger sample size ( $N$ ) before the termination criteria of our SAA algorithm are met.

## 5.6 Evaluating the SAA Framework

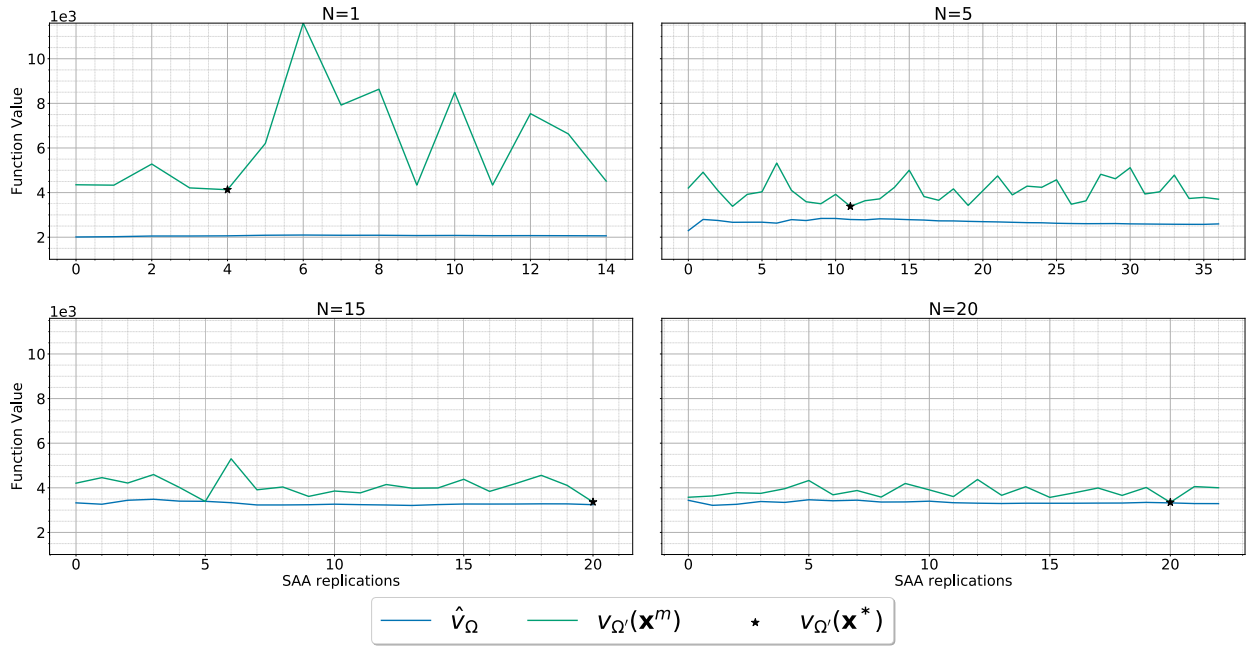
A common problem in the implementation of SAA algorithms is selecting the parameters  $N$  (number of scenarios in set  $\Omega$ ) and  $M$  (number of SAA replications). Both parameters are instance-dependent: low values of  $M$  and  $N$  lead to poor solutions, whereas high values of  $M$  and  $N$  significantly increase computation times. In this work, we aim to avoid this problem through an incremental update scheme for  $N$  and a termination criteria based on a gap estimate  $\epsilon_m$  (Algorithm 1), instead of a fixed number of iterations  $M$ . In this section, we conclude with an empirical evaluation of these design choices.

To establish the impact of our incremental update scheme, we solve the benchmark instances (Table 2) for fixed values of  $N$  by setting  $\Delta = 0$  in Algorithm 1. We terminate SAA as soon as  $\sigma_{\hat{v}_\Omega}$  is less than 0.01 (line 15 in Algorithm 1). For these experiments, we use recourse policy  $R_0$  and  $\hat{c}_v^2 = 0.25$ .

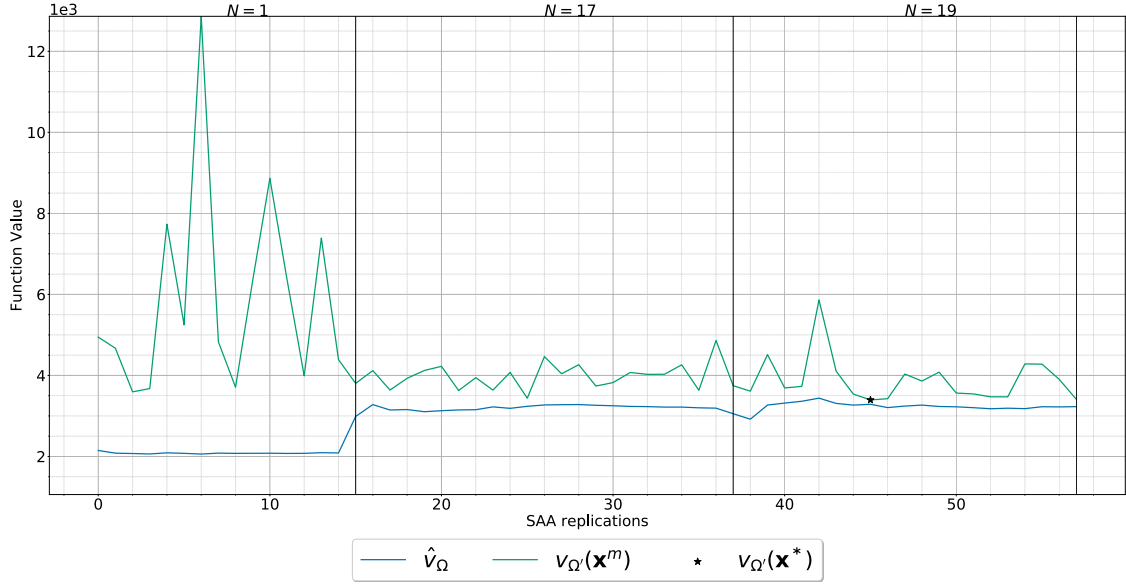
As an example, we solve instance  $I_5$  for fixed  $N = 1, 5, 15, 20$  with our incremental update scheme. The results are depicted in Figure 8a. For each SAA replication  $m$  (x-axis) the graphs show the value  $\hat{v}_\Omega$  and the objective value  $v_{\Omega'}(\mathbf{x}^m)$ . Clearly, when  $N$  is too small (e.g.  $N = 1$ ), the objective value  $v_{\Omega'}(\mathbf{x}^m)$  fluctuates considerably. For larger values of  $N$  the objective value becomes more stable, indicating that the solution  $\mathbf{x}^m$  obtained by solving the  $m^{th}$  SAA replication with  $|\Omega| = N$  closely approximates the value  $v_{\Omega'}(\mathbf{x}^m)$ . Figure 8b shows that our incremental scheme increases the value of  $N$  twice, and terminates after  $m = 60$  iterations. Here, the best solution value is found for  $N = 19$ .

Table 4 summarizes the results, averaged over *all* instances with the same number of customers  $|C|$ . Similar to the experiments in the previous sections, the SAA algorithm has been invoked 3 times per instance. For a given  $|C|$ , Table 4 reports the average results obtained with fixed  $N = 1, 5, 15, 30, 60, 90$ , and with the proposed incremental update scheme ( $\sim$ ). For a given  $|C|$  and  $N$ , we state the best objective value found  $v_{\Omega'}(\mathbf{x}^*)$  by the time the algorithm terminates, the time at which this best solution was discovered (TTB), the total execution time (T(s)), and the number of times ( $\mathcal{N}$ ) SAA terminated with  $\epsilon_m \leq 0.05$ . Here, we remind the reader that when  $\epsilon_m > 0.05$  our incremental update scheme would increase sample size  $N$  (line 19 in Algorithm 1).

Observe from Table 4 that when  $|C|$  equals 15 and 20, the best results are obtained for  $N = 30$ , whereas for  $|C| = 30$ , the best result is found for  $N = 90$ . Manually picking the ideal value of  $N$  is hard without enumerating different values of  $N$ . Fortunately our incremental update scheme ( $\sim$ ) finds solutions of comparable quality, without the need to manually select



(a) SAA replications solved with a fixed sample size  $N = 1, 5, 15, 20$ .



(b) SAA replications with sample sizes adjusted throughout the method.

Figure 8: Comparison of the evolution of the SAA framework considering fixed sample sizes  $N$  (a) and the adjusted sizes (b)

$N$ , and relatively fast when compared against the computation times for larger  $N$ .

## 6 Conclusions

We addressed a stochastic variant to a last-mile delivery problem, considering trunk delivery. In this case, a customer's car is used to facilitate the delivery process of direct-to-consumer orders.



$N$	$ C  = 15$				$ C  = 20$			
	$v_{\Omega'}(\mathbf{x}^*)$	$\mathcal{N}$	TTB	T(s)	$v_{\Omega'}(\mathbf{x}^*)$	$\mathcal{N}$	TTB	T(s)
1	4,108	0	15	32	5,182	0	23	41
5	3,528	0	24	70	4,039	1	54	108
15	3,453	0	58	136	3,915	1	122	218
30	3,414	3	70	215	3,829	1	108	383
60	3,427	3	145	463	3,895	3	783	994
90	3,434	5 <sup>a</sup>	175	816	3,933	5 <sup>a</sup>	397	1,301
~	3,444	5 <sup>a</sup>	108	297	3,834	5 <sup>a</sup>	445	994

$ C  = 30$				
	$v_{\Omega'}(\mathbf{x}^*)$	$\mathcal{N}$	TTB	T(s)
1	9,212	0	62	95
5	7,304	1	108	260
15	6,878	0	360	582
30	6,796	5	680	1,174
60	6,785	8	1,081	2,177
90	6,774	10 <sup>a</sup>	2,396	4,737
~	6,796	10 <sup>a</sup>	2,143	3,373

Table 4: Tuning the SAA with a fixed  $N$ .  $v_{\Omega'}(\mathbf{x}^*)$ : best solution found;  $\mathcal{N}$ : the number of times SAA terminated with  $\epsilon_m \leq 0.05$ ; TTB: time when best solution found; T(s): total execution time.  $\hat{c}_v^2 = 0.25$  and  $R_0$  as second-stage recourse action. <sup>a</sup>All SAA invocations terminated with  $\epsilon_m \leq 0.05$ .

This problem is modeled as a Vehicle Routing Problem with Roaming Delivery Locations. Customers are assumed to be available for delivery at different locations as he/she moves along an itinerary, defining non-overlapping windows for servicing. By considering uncertain travel times when designing the routes for service, not only delivery vehicles are affected by possible differences between expected and realized times, but also customers, since travel times also affect the availability of customers at each location along their itinerary. Moreover, the presence of different locations for delivery to a customer might provide service providers with new approaches to handle uncertain events (e.g., road disruptions) while maintaining a desirable service level to customers.

We tackled the problem as a two-stage stochastic problem and implemented a hierarchical, scenario-based sample approximation method, in combination with an adaptation of local search heuristics, to take travel time uncertainty into account. Experiments conducted on a set of VRPDL instances showed that planning the delivery routes while explicitly taking stochasticity into account leads to significant savings compared to a deterministic approach which solely considers expected travel time values. Routing plans obtained by using expected values are too conservative when travel time realizations are shorter than expectations, or induce many missed deliveries in the advent of significant traffic disruptions. The SAA approach used in this paper resolves this issue by optimizing routes over a large number of potential travel time realizations.

Future research could involve the role of new technology (e.g., machine learning) to monitor and predict daily itineraries of receivers and drivers. This could allow, for example, delivery routes being adjusted dynamically, using real-time information provided by customers given the current status of the network.

## Acknowledgments

This research was supported in part by the Netherlands Organisation for Scientific Research (NWO) grant 629.002.211 and the Dutch Institute for Advanced Logistics grant 439.17.551.

## References

- Adulyasak, Y. and Jaillet, P. (2016). Models and algorithms for stochastic and robust vehicle routing with deadlines. *Transportation Science*, 50(2):608–626.
- Bates, J. and Leibling, D. (2012). Spaced out: Perspectives on parking policy. RAC Foundation. <https://tinyurl.com/7vn8gcw>. Accessed 20-January-2019.
- Bektaş, T., Erdoğan, G., and Røpke, S. (2011). Formulations and branch-and-cut algorithms for the generalized vehicle routing problem. *Transportation Science*, 45(3):299–316.
- Biesinger, B., Hu, B., and Raidl, G. (2016). An integer L-shaped method for the generalized vehicle routing problem with stochastic demands. *Electronic Notes in Discrete Mathematics*, 52:245 – 252. INOC 2015 – 7th International Network Optimization Conference.
- Binart, S., Dejax, P., Gendreau, M., and Semet, F. (2016). A 2-stage method for a field service routing problem with stochastic travel and service times. *Computers & Operations Research*, 65:64 – 75.

705 Branda, M. (2012). Sample approximation technique for mixed-integer stochastic programming  
706 problems with several chance constraints. *Operations Research Letters*, 40(3):207 – 211.

707 Ehmke, J. F., Campbell, A. M., and Urban, T. L. (2015). Ensuring service levels in routing  
708 problems with time windows and stochastic travel times. *European Journal of Operational  
709 Research*, 240(2):539 – 550.

710 Errico, F., Desaulniers, G., Gendreau, M., Rei, W., and Rousseau, L.-M. (2016). A priori  
711 optimization with recourse for the vehicle routing problem with hard time windows and  
712 stochastic service times. *European Journal of Operational Research*, 249(1):55 – 66.

713 Gendreau, M., Ghiani, G., and Guerriero, E. (2015). Time-dependent routing problems: A  
714 review. *Computers & Operations Research*, 64:189 – 197.

715 Gendreau, M., Jabali, O., and Rei, W. (2016). 50th anniversary invited article—future research  
716 directions in stochastic vehicle routing. *Transportation Science*, 50(4):1163–1173.

717 Ghiani, G. and Improta, G. (2000). An efficient transformation of the generalized vehicle  
718 routing problem. *European Journal of Operational Research*, 122(1):11 – 17.

719 Hawkins, A. (2018). Amazon will now deliver packages to the trunk of your car. Verge.  
720 <https://tinyurl.com/yaektvtd>. Accessed 20-January-2019.

721 Hvattum, L. M., Løkketangen, A., and Laporte, G. (2006). Solving a dynamic and stochastic  
722 vehicle routing problem with a sample scenario hedging heuristic. *Transportation Science*,  
723 40(4):421–438.

724 Jabali, O., Leus, R., Van Woensel, T., and de Kok, T. (2015). Self-imposed time windows in  
725 vehicle routing problems. *OR Spectrum*, 37(2):331–352.

726 Kleywegt, A., Shapiro, A., and Homem-de Mello, T. (2002). The sample average approximation  
727 method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502.

728 Kovacs, A. A., Golden, B. L., Hartl, R. F., and Parragh, S. N. (2015). The generalized consistent  
729 vehicle routing problem. *Transportation Science*, 49(4):796–816.

730 Laporte, G. and Louveaux, F. V. (1993). The integer L-shaped method for stochastic integer  
731 programs with complete recourse. *Operations Research Letters*, 13(3):133 – 142.

732 Laporte, G., Louveaux, F. V., and van Hamme, L. (2002). An integer L-shaped algorithm  
733 for the capacitated vehicle routing problem with stochastic demands. *Operations Research*,  
734 50(3):415–423.

735 Li, X., Tian, P., and Leung, S. C. (2010). Vehicle routing problems with time windows and  
736 stochastic travel and service times: Models and algorithm. *International Journal of Produc-  
737 tion Economics*, 125(1):137 – 145.

738 Lombard, A., Tamayo-Giraldo, S., and Fontane, F. (2018). Vehicle routing problem with  
739 roaming delivery locations and stochastic travel times (VRPRDL-S). *Transportation Research  
740 Procedia*, 30:167 – 177. EURO Mini Conference on Advances in Freight Transportation and  
741 Logistics.

- 742 Miranda, D. M. and Conceição, S. V. (2016). The vehicle routing problem with hard time  
743 windows and stochastic travel and service time. *Expert Systems with Applications*, 64:104 –  
744 116.
- 745 Moccia, L., Cordeau, J.-F., and Laporte, G. (2012). An incremental tabu search heuristic  
746 for the generalized vehicle routing problem with time windows. *Journal of the Operational  
747 Research Society*, 63(2):232–244.
- 748 Ozbaygin, G., Karasan, O. E., Savelsbergh, M., and Yaman, H. (2017). A branch-and-price  
749 algorithm for the vehicle routing problem with roaming delivery locations. *Transportation  
750 Research Part B: Methodological*, 100:115 – 137.
- 751 Reyes, D., Savelsbergh, M., and Toriello, A. (2017). Vehicle routing with roaming delivery  
752 locations. *Transportation Research Part C: Emerging Technologies*, 80:71 – 91.
- 753 Russell, R. A. and Urban, T. L. (2008). Vehicle routing with soft time windows and erlang  
754 travel times. *Journal of the Operational Research Society*, 59(9):1220–1228.
- 755 Schultz, R., Stougie, L., and van der Vlerk, M. H. (1998). Solving stochastic programs with inte-  
756 ger recourse by enumeration: A framework using gröbner basis. *Mathematical Programming*,  
757 83(1):229–252.
- 758 Taş, D., Dellaert, N., van Woensel, T., and de Kok, T. (2013). Vehicle routing problem  
759 with stochastic travel times including soft time windows and service costs. *Computers &  
760 Operations Research*, 40(1):214 – 224.
- 761 Taş, D., Dellaert, N., van Woensel, T., and de Kok, T. (2014a). The time-dependent vehicle  
762 routing problem with soft time windows and stochastic travel times. *Transportation Research  
763 Part C: Emerging Technologies*, 48:66 – 83.
- 764 Taş, D., Gendreau, M., Dellaert, N., van Woensel, T., and de Kok, A. (2014b). Vehicle routing  
765 with soft time windows and stochastic travel times: A column generation and branch-and-  
766 price solution approach. *European Journal of Operational Research*, 236(3):789 – 799.
- 767 Tom Tom (2016). Tomtom traffic index: Measuring congestion worldwide. Tom Tom Traffic  
768 Index. <https://tinyurl.com/y6wgf8e6>. Accessed 20-January-2019.
- 769 Toth, P. and Vigo, D. (2014). *Vehicle Routing: Problems, Methods, and Applications, Second  
770 Edition*. Number 18 in MOS-SIAM Series on Optimization. SIAM.
- 771 Vareias, A. D., Repoussis, P. P., and Tarantilis, C. D. (2017). Assessing customer service  
772 reliability in route planning with self-imposed time windows and stochastic travel times.  
773 *Transportation Science*, Articles in Advance(on-line):1–26.
- 774 Verweij, B., Ahmed, S., Kleywegt, A. J., Nemhauser, G., and Shapiro, A. (2003). The sample  
775 average approximation method applied to stochastic routing problems: A computational  
776 study. *Computational Optimization and Applications*, 24(2):289–333.

## Appendix A Linear constraints (19)

Let  $r_c$ ,  $c \in C$ , be an integer, uniquely identifying the route servicing customer  $c$  e.g.,  $r_c = r_{c'}$  if and only if  $c$  and  $c'$  are serviced by the same route. Consider the following set of constraints:

$$x_{ij}^+(\omega) \leq 1 - \min\{1, |r_{\pi(i)} - r_{\pi(j)}|\} \quad \forall(i, j) \in A_R, i \neq 0, j \neq n+1 \quad (30)$$

$$|r_c - r_{c'}| \geq 1 - |C|(2 - \sum_{i \in N_c} x_{0i} - \sum_{j \in N_{c'}} x_{0j}) \quad \forall c, c' \quad (31)$$

$$r_c \leq \sum_{j \in N} x_{0j} \quad \forall c \in C \quad (32)$$

$$r_c \geq r_{c'} - |C|(1 - \sum_{i \in N_c} \sum_{j \in N_{c'}} x_{ij}) \quad \forall c, c' \in C \quad (33)$$

$$r_{c'} \geq r_c - |C|(1 - \sum_{i \in N_c} \sum_{j \in N_{c'}} x_{ij}) \quad \forall c, c' \in C \quad (34)$$

$$r_c \in \mathbb{Z}^+ \cup \{0\} \quad \forall c \in C \quad (35)$$

Inequalities (30) impose that a second-stage rerouting decision  $x_{ij}^+(\omega)$  is only allowed between locations  $i, j$  corresponding to customers serviced by the same route (observe that if customers  $c, c'$  have  $r_c \neq r_{c'}$  then  $x_{ij}^+(\omega) \leq 0$  by (30)). Inequalities (31) state that if  $c$  is the first customer visited in a route and  $c'$  is also the first customer visited in another route then  $r_c$  and  $r_{c'}$  should be assigned different values. Constraints (32) limit the assigned value for any  $r_c$  to at most the number of routes in the first-stage solution. Inequalities (33) and (34) state that whenever customer  $c'$  is visited immediately after  $c$  in a given route, these customers are serviced by the same route and, thus,  $r_c = r_{c'}$ .

Let  $y_{c,c'} = |r_c - r_{c'}|$  and  $s_{c,c'}^1, s_{c,c'}^2$  binary variables  $\forall c, c' \in C$ , then:

$$0 \leq y_{c,c'} - (r_c - r_{c'}) \leq 2|C|s_{c,c'}^1 \quad (36)$$

$$0 \leq y_{c,c'} - (r_{c'} - r_c) \leq 2|C|s_{c,c'}^2 \quad (37)$$

$$s_{c,c'}^1 + s_{c,c'}^2 = 1 \quad (38)$$

Moreover, if  $z_{c,c'} = \min\{1, y_{c,c'}\}$ , then:

$$z_{c,c'} \leq 1 \quad (39)$$

$$z_{c,c'} \leq y_{c,c'} \quad (40)$$

$$z_{c,c'} \geq t_{c,c'}^1 \quad (41)$$

$$z_{c,c'} \geq y_{c,c'} - |C|(1 - t_{c,c'}^2) \quad (42)$$

$$t_{c,c'}^1 + t_{c,c'}^2 = 1 \quad (43)$$

We now replace (30) and (31) by:

$$x_{ij}^+(\omega) \leq 1 - z_{\pi(i), \pi(j)} \quad \forall(i, j) \in A_R, i \neq 0, j \neq n+1 \quad (44)$$

$$y_{c,c'} \geq 1 - |C|(2 - \sum_{i \in N_c} x_{0i} - \sum_{j \in N_{c'}} x_{0j}) \quad \forall c, c' \quad (45)$$

respectively. Inequalities (19)-(20) and variables  $u_{c,c'}$  in the two-stage model are replaced by the set of inequalities (32)-(45) and by variables  $y_{c,c'}$  and  $z_{c,c'}$ .