Published in "Computers & Operations Research", 2020, vol. 113, which should be cited to refer to this work.
DOI:10.1016/j.cor.2019.104798

Inventory routing with non-stationary stochastic demands

Iliya Markov * Michel Bierlaire * Jean-François Cordeau †
Yousef Maknoon * Sacha Varone ‡

April 26, 2017

Report TRANSP-OR 160825
Transport and Mobility Laboratory
École Polytechnique Fédérale de Lausanne
transp-or.epfl.ch

*Transport and Mobility Laboratory, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland, {iliya.markov@epfl.ch, michel.bierlaire@epfl.ch, yousef.maknoon@epfl.ch}

†CIRRELT and HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7, jean-francois.cordeau@hec.ca

‡Haute École de Gestion de Genève, University of Applied Sciences Western Switzerland (HES-SO), CH-1227 Carouge, Switzerland, sacha.varone@hesge.ch

Abstract

We solve a rich logistical problem inspired from practice, in which a heterogeneous fixed fleet of vehicles is used for collecting recyclable waste from large containers over a finite planning horizon. Each container is equipped with a sensor, which communicates its level at the start of the day. Given a history of observations, a forecasting model is used to estimate the point demand forecasts as well as a forecasting error representing the level of uncertainty. The problem falls under the framework of the stochastic inventory routing problem. We introduce dynamic probabilistic information in the solution process, which impacts the cost through the probability of container overflows on future days and the probability of route failures. We cast the problem as a mixed integer non-linear program and, to solve it, we implement an adaptive large neighborhood search algorithm, which integrates a specialized forecasting model, tested and validated on real data. Computational testing demonstrates that our algorithm performs very well on inventory routing and vehicle routing benchmarks from the literature. We are able to evaluate the benefit of including uncertainty in the objective function on rich IRP instances derived from real data coming from the canton of Geneva, Switzerland. Our approach performs significantly better compared to alternative policies in its ability to limit the occurrence of container overflows for the same routing cost. We also analyze the solution properties of a rolling horizon approach and derive empirical lower and upper bounds.

Keywords: stochastic inventory routing problem; demand forecasting; adaptive large neighborhood search; uncertainty; waste collection

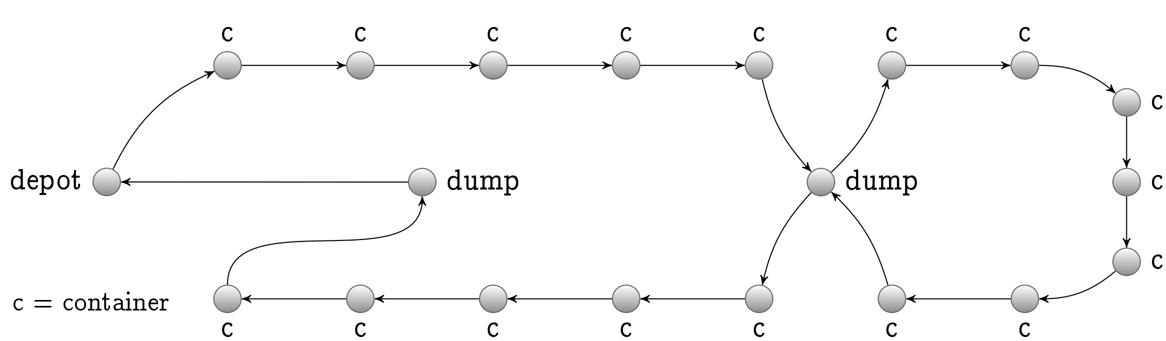
1 Introduction

Waste collection is one of the most important logistical activities performed by any municipality, and also one of the most expensive. According to various estimates, collection costs account for more than 70% of waste management costs (Johansson, 2006; Tavares et al., 2009; Greco et al., 2015). Recycling, on the other hand, can alleviate problems related to landfill capacity and pollution, and many countries have already set ambitious target levels for recycling. As part of its Circular Economy Strategy, the European Union (EU), for example, has adopted legislative proposals to set a common EU target for recycling 65% of municipal and 75% of packaging waste by 2030, limiting at the same time the use of landfills (European Commission, 2016). Given the high cost of waste management and the significant proportion of collection costs, even small improvements in the latter can lead to substantial financial savings for waste collectors, municipalities, and ultimately the taxpayer.

In this context, we solve a rich recyclable waste collection problem, which can be described as follows. A heterogeneous fixed fleet of vehicles with different speeds, capacities, fixed and variable costs, is used for collecting recyclable waste over a finite planning horizon, say a week to 10 days. Since both waste containers and collection vehicles are flow-specific, the problem can be decomposed and solved separately for each waste flow. As shown in Figure 1, each tour starts and ends at the depot, and is a sequence of collections followed by disposals at the available dumps. All collections are of the same waste flow and all visited dumps accept the latter. There is a mandatory visit to a dump just before the end of a tour, i.e. a tour terminates with an empty vehicle. Dumps are recycling plants. There could be multiple dumps for the collected waste flow and they can be used when and as needed along the tour. We consider time windows for the depots, containers and dumps. A tour is also limited by the legal duration of the working day. Accessibility restrictions apply to certain points, for example for containers located in narrow streets that cannot be accessed by big collector trucks.

Each container is equipped with an ultrasound sensor that communicates the waste level to a central database at the start of each day. Given the availability of historical data, a statistical model is used to estimate for each container the point demand forecasts for each day of the planning horizon. In addition, the model fit gives a consistent estimate of the forecasting

Figure 1: Example of a Collection Tour (Markov et al., 2016)



error, which is used to calculate the risk of container overflows and route failures during the planning horizon. Experience suggests that full containers continue serving demand because people place the waste beside them. Nonetheless, if a sensor communicates that a container is full, it must be collected within the day. The collection policy thus assumes a back-ordering inventory decision, because the collector is charged a penalty for a full container, but the number of back-order days is limited to one.

Given the multi-day planning horizon, our problem falls under the framework of the Stochastic Inventory Routing Problem (SIRP), with no inventory holding costs and unlimited inventory capacity at the dumps. As a counterpart to inventory holding costs, we consider container overflow costs, which the collector pays to the municipality in the occurrence of such events. Their correct attribution to the objective function involves the calculation of conditional probabilities, which are day-dependent and dynamically affected by previous collections during the planning horizon, and which lead to non-linearities. The contribution of our research is fourfold. First, we incorporate dynamic probabilistic information in the solution process, which impacts the cost through the probability of container overflow on future days and the probability of route failures, measured by the likelihood that a vehicle does not have sufficient capacity to serve the realized demands before its next scheduled dump visit. Secondly, we utilize a demand forecasting model that has been specifically designed for our purpose, and tested and validated on real data, and integrate it with a state-of-the-art Adaptive Large Neighborhood Search (ALNS) algorithm. Thirdly, our embedded vehicle routing problem (VRP) contains a variety of rich features traditionally absent or rarely considered in the literature on the Inventory Routing Problem (IRP), such as a heterogeneous fixed fleet, intermediate facilities and time windows. Fourthly, the extensive computational testing demonstrates that our algorithm produces excellent results on IRP benchmark and very good results on VRP benchmarks from the literature. We are able to evaluate the benefit of considering uncertainty in the objective function on a set of rich IRP instances derived from real data coming from the canton of Geneva, Switzerland. The SIRP performs significantly better than alternative practical policies in its ability to control the occurrence of container overflows for the same routing cost. We also analyze the solution properties of a rolling horizon approach for a dynamic and stochastic version of the problem and derive empirical lower and upper bounds on its solution cost.

The remainder of this article is organized as follows. Section 2 positions our work with respect to the relevant VRP and IRP literature. Section 3 outlines the forecasting model and formalizes our SIRP with a mathematical formulation. Section 4 describes the ALNS algorithm and the solution methodology. Section 5 presents the numerical experiments, and finally Section 6 concludes and explores future work directions.

2 Related Literature

Given the rich features of our problem, we position our contribution both within the VRP and the IRP literature. In Section 2.1 below, we conduct a short survey of the VRP with Intermediate Facilities (VRP-IF), the electric and alternative fuel VRP, and the heterogeneous fixed fleet VRP. Afterwards, in Section 2.2, we shift our attention to the stochastic IRP with a specific focus on the modeling approach with respect to the treatment of uncertainty.

2.1 Related VRP Literature

One of the seminal applications of the VRP to waste collection is that of Beltrami and Bodin (1974), who solve a periodic VRP-IF for commercial waste collection in New York City. Angelelli and Speranza (2002b) apply Cordeau et al.'s (1997) tabu search heuristic to a similar periodic problem. The latter's methodological framework is used by Angelelli and Speranza (2002a) to analyze the operational cost benefits of several different collection policies in Val Trompia, Italy and Antwerp, Belgium. Intermediate facilities in a distribution context are used by Bard et al. (1998a) who develop a branch-and-cut algorithm and use it for instances of limited size.

Kim et al. (2006) solve the waste collection VRP-IF by simulated annealing, explicitly considering also features such as tour compactness and workload balancing. A related problem, the Multi-depot VRP with Inter-depot routes (MDVRPI), is proposed by Crevier et al. (2007). They use the adaptive memory principle of Rochat and Taillard (1995) and decompose the problem into multi-depot, single-depot and inter-depot subproblems, which are solved using tabu search. A solution to the MDVRPI is obtained through a set covering formulation and improved by tabu search. Crevier et al. (2007) generate two benchmark sets with a fixed homogeneous fleet stationed at one depot, with the rest of the depots acting only as intermediate facilities. Muter et al. (2014) develop a branch-and-price algorithm for the MDVRPI and manage to solve to optimality some instances with up to 50 customers.

A conceptually similar problem appears in the routing of electric and alternative fuel vehicles, where recharging or refueling decisions correspond to emptying decisions. Conrad and Figliozzi (2011) consider the recharging VRP, where electric vehicles can recharge at customer locations with time windows. Erdoğan and Miller-Hooks (2012) treat the green VRP, where vehicles use a sparse alternative fuel infrastructure. Schneider et al. (2014) solve the electric VRP with time windows and recharging stations, while Schneider et al. (2015) combine recharging and reloading facilities in the VRP with intermediate stops. A survey of the relevant literature is available in Moghaddam (2015) and Pelletier et al. (2014).

The preceding literature assumes homogeneous fleets, whether limited or not. However, in industry fleets are rarely homogeneous. They either start as heterogeneous or become such as vehicles are added or replaced. More recently, Hiermann et al. (2014), Sassi et al. (2014), Goeke and Schneider (2015), and Mancini (2015) have started filling the gap by considering conventional and alternative fuel vehicles simultaneously. Taillard (1999) was the first to formally define the Heterogeneous Fixed Fleet VRP (HFFVRP). Being a generalization of the Vehicle Fleet Mix Problem (VFMP), the HFFVRP is more difficult than the classical VRP or the VFMP. Taillard's (1999) solution approach relies on heuristic column generation with AM, and vehicle assignment costs are calculated at each iteration. The best heuristic approaches for this problem are due to Penna et al. (2013) and Subramanian et al. (2012), and the only fully exact method is that of Baldacci and Mingozzi (2009).

The vehicle routing subproblem embedded in our IRP already includes most of the features discussed above, notably a heterogeneous fixed fleet, multiple dumps playing the role of intermediate facilities, in addition to time windows, a maximum tour duration, and accessibility restrictions. The simultaneous presence of all these features is seldom considered in the VRP literature. Our problem has the complication of including them in an IRP context. Thus, while they are essential to describing a realistic problem inspired from practice, they also pose a great challenge in terms of modeling and solution methodology.

2.2 Related SIRP Literature

Coelho et al. (2014b) conduct a survey of the IRP literature during the past thirty years. Table 1 positions our problem in terms of the structural classification scheme they propose. We consider a finite planning horizon which is used in a rolling fashion. There are multiple containers that are emptied into multiple dumps, and so we identify the structure as many-to-many. Multiple containers can be visited along a tour and the inventory policy is Order-Up-to (OU), meaning that a visited container is always fully emptied. Container overflow is served at a penalty (back-order) and there is a limit on the number of back-order days. The fleet is heterogeneous and fixed. Information-wise, the problem is stochastic and, when solved in a rolling horizon fashion, dynamic with new container information revealed each day. Other comprehensive surveys and literature reviews on the IRP are available in Abdelmaguid (2004), Moin and Salhi (2007), Andersson et al. (2010), Yu et al. (2012), Coelho et al. (2014b), Ivarsøy and Solhaug (2014) and Park et al. (2016), and a particular focus on stochastic problems and aspects can be found in Moin and Salhi (2007), Yu et al. (2012) and Coelho et al. (2014b). In the following, we limit our attention to finite-horizon stochastic problems, i.e. the class to which our problem belongs. In particular, we emphasize on the use of a rolling horizon approach, the limitations of relying on the concept of optimal service frequencies, and the pros and cons of various modeling approaches with respect to the problem's stochastic elements.

Trudeau and Dror (1992) extend the work of Dror and Ball (1987) on the optimal service

Table 1: Structural Classification (Coelho et al., 2014b)

Criterion	Classification
Time horizon	Finite (rolling)
Structure	Many-to-many
Routing	Multiple
Inventory policy	Order-Up-to (OU)
Inventory decisions	Back-ordering (with a penalty and limit)
Fleet composition	Heterogeneous
Fleet size	Multiple (fixed)

frequency under a stochastic setting. They consider both stock-outs and route failures. Unlike previous research (see e.g. Stewart and Golden, 1983; Dror et al., 1985; Dror and Levy, 1986; Dror and Ball, 1987; Larson, 1988) which uses a vehicle with an artificially small capacity to avoid route failures, Trudeau and Dror (1992) develop an analytical probability expression. Our work differs from that of Trudeau and Dror (1992) in several major aspects. In particular, we have a heterogeneous fixed fleet. Route failure in our case applies to a depot-to-dump or a dump-to-dump trip, of which there could be several in a given tour. Finally, we do not impose a maximum of one visit and one overflow per container during the planning horizon, which precludes the derivation of an exact closed-form probability measure. On the contrary, it requires the complicated management of binary trees, tracking each container's visit-dependent and conditional probability of overflow on each day of the planning horizon. In addition, we consider multiple rich routing features.

The work of Bard et al. (1998b) includes intermediate facilities in a distribution context. They apply problem decomposition with a two-week rolling horizon. Customers to be visited during the planning horizon are identified and those scheduled for the first week are routed, after which the horizon is rolled over by a week. The customer selection procedure is based on Jaillet et al. (2002) who derive the optimal restocking frequency and the incremental cost of deviating from it. In the first step of the decomposition scheme, customers whose optimal visit day falls within the two-week horizon are assigned to specific days by solving a balanced generalized assignment problem that minimizes the total incremental cost, accounting for uncertainty through a lower and upper bound on the total daily demand to be served. The solution of the routing problem relies on construction and improvement heuristics including inter-day customer exchanges. Similar ideas, based on the identification of customers who must be served versus those who may be served are used in Bitsch (2012) and Mes et al. (2014), both with applications to waste collection where the objective is the minimization of overflows. The former relies on the calculation of incremental costs, while the latter on expectation-based service frequency. Due to the implied repetitive pattern, this type of approaches is only appropriate in situations where demand stationarity can be assumed.

Campbell and Savelsbergh (2004) also deal with uncertainty through a decomposition approach that solves the problem of assigning customers to days first, using the cost of a giant TSP tour as a crude measure of the daily routing cost, and with coarser period aggregations toward the end of the planning horizon. Afterwards, the IRP is solved for the first few days of the planning horizon for the customers that were assigned there and assuming deterministic information. This approach is used in a rolling horizon framework with the benefit of reflecting longer-term costs in the shorter-term problem, i.e. on the days for which the actual IRP is solved. Such a balance, usually expressed through a so-called reduction procedure, was the focus of much of the above-mentioned IRP research (see Dror and Ball, 1987; Trudeau and Dror, 1992; Dror and Trudeau, 1996; Jaillet et al., 2002). Stochasticity is also discussed in Coelho et al. (2014a), who present a modeling and solution framework for dynamic and stochastic IRP, incorporating the use of forecasting. However, their approach relies on constructing point forecasts to be used in a rolling horizon fashion without explicit incorporation of probabilistic information in the solution process. Independent of the modeling approach or the methodology used, the rolling horizon technique is useful in dealing with uncertainty by helping make forward-looking decisions in the operational short-term.

More recently, research on the SIRP has dealt with uncertainty in various ways. Solyalı et al. (2012), for example, use the robust optimization approach introduced by Bertsimas and Sim (2003, 2004) to solve a problem with dynamic uncertain demands, ensuring that vehicle capacity will not be violated for any realization of the customer demands, which are independent and symmetric, and for which only a point estimate and a maximum deviation are specified. They develop a strong formulation and use a branch-and-cut solution approach. Bertazzi et al. (2013) propose a heuristic rollout algorithm that uses a sampling approach to generate demand scenarios for the current period and considers the average demand for future ones. Decisions are made by solving a mixed integer program by branch-and-cut in each period. A similar approach is used by Bertazzi et al. (2015) who apply it to an IRP with transportation procurement. Adulyasak et al. (2015) propose a two-stage and a multi-stage approach for a production-routing problem under demand uncertainty, in which the first stage determines production setup and visit frequencies, while subsequent stages determine production and delivery quantities. They develop exact formulations and a branch-and-cut algorithm, and for handling a large number of scenarios, they propose a Benders decomposition approach, which is able to solve instances of realistic size. Stochastic optimization with recourse is used by Hemmelmayr et al. (2010) and Nolz et al. (2014), who present applications related to blood product distribution and medical waste collection, respectively. Chance-constrained approaches, often oriented towards maintaining a service level, can be found in Yu et al. (2012), Abdollahi et al. (2014), Soysal et al. (2015) and Soysal et al. (2016), while static risk expressions in the objective function that use the demand distribution parameters are applied by Nekooghadirli et al. (2014a) and Nekooghadirli et al. (2014b).

The use of a particular modeling approach has a strong influence on how the problem at hand is being viewed. Robust optimization, for example, protects against the worst case scenario for a given budget of uncertainty. Thus, it has a clear risk-aversion bias. However, it still leaves the question of how to define an appropriate budget of uncertainty. And more generally, this approach is less relevant for our problem where container overflows and route failures are not disastrous events. Their states are frequently revisited, unlike what is usually the case in robust optimization. Furthermore, container overflows and route failures have a monetary cost which should figure in the total expected cost incurred by the collector. Thus, the integration of probability information in the objective is used to provide a monetary dimension to these stochastic events, and this approach has a clear cost bias, as would be the case for a cost-minimizing firm. Scenario generation and chance-constrained approaches fall in the middle. While scenario generation/stochastic programming would be very cumbersome computationally for a rich IRP like ours, chance constraints may be integrated in our approach, although the value added would probably be minimal, given that we can control the degree of conservatism of the occurrence of stochastic events by modifying their associated costs or penalties in the objective function. Our IRP has a cost bias and we include rich probability information in the objective function. Moreover, unlike previous IRP research, we do not assume a stationary demand distribution and, therefore, cannot rely on the estimation of optimal service frequencies or cyclic schedules such as in a periodic VRP. In terms of calculating the overflow probabilities, there are certain similarities to the conditional calculation of stock-outs used in Ribeiro and Lourenço (2003). However, they formulate unrealistic assumptions on the ability to observe inventory in future periods, and their problem is finally solved using a simple heuristic approach.

3 Formulation

In what follows, Section 3.1 presents a brief sketch of the forecasting model, Section 3.2 develops a mathematical formulation for our SIRP, and Section 3.3 discusses the necessary changes to the latter for solving benchmark instances from the literature. Table 2 summarizes the used notations. Some of the notations, in particular the inventory holding cost, are only used in the model reformulations presented in Section 3.3 but are still included in the table for completeness and ease of reference. We note that container demand refers to the volume amount placed in a container on a given day. Container inventory and capacity are also measured in terms of volume. Vehicles, on the other hand, have both volume and weight capacities. Depending on the density of the collected waste flow, one of them becomes limiting while the other may not be. However, we observe that if the weight capacity becomes limiting before the volume capacity, the volume capacity can be adjusted to become limiting at the same time. Through this simple preprocessing step, we avoid tracking both volume and weight for the benefit of a more elegant formulation.

3.1 Forecasting Model

Markov et al. (2015) propose a forecasting model for the daily container demands which exhibits superior in- and out-of-sample performance compared to alternatives. It is based on a discrete mixture of count-data models describing populations depositing different waste volumes in the containers. Thus, it supposedly captures a realistic underlying behavior though simplified. We assume a set $\mathcal{V}$ of distinct deposit volumes, where deposit volume $\nu \in \mathcal{V}$ is generated with a Poisson rate $\xi_{it\nu}$ for container i on day t. The rate $\xi_{it\nu}$ takes the functional form $\xi_{it\nu} = \exp(\boldsymbol{\kappa}_{it}^{\top}\boldsymbol{\gamma}_{\nu})$, where $\boldsymbol{\kappa}_{it}$ is a vectors of covariates, such as the day of the week, weather variables, holiday periods, etc., and $\boldsymbol{\gamma}_{\nu}$ is a vector of estimable parameters for deposit volume ν. We formulate an expression for the expected value of the demand of container i on day t as follows:

$$\mathbb{E}\left(\rho_{it}\right) = \sum_{\nu \in \mathcal{V}} \nu \xi_{it\nu}. \tag{1}$$

To fit the model, we minimize the sum of squared errors between the observed ρ_{it}^{o} and the expected demand $\mathbb{E}\left(\rho_{it}\right)$ over the set of containers $\mathcal{P}$ and a historical period $\mathcal{H}$ of data avail-

Table 2: Notations

Sets			
$\mathcal{V}$	set of container deposit volumes	$\mathcal{H}$	historical estimation period
o	origin	d	destination
$\mathcal{D}$	set of dumps	$\mathcal{P}$	set of containers
$\mathcal{N}$	set of all points $= \{o\} \cup \{d\} \cup \mathcal{D} \cup \mathcal{P}$	$\mathcal{K}$	set of vehicles
$\mathcal{T}$	planning horizon $= \{0, \dots, u\}$	$\mathcal{T}^+$	shifted horizon $= \{1, \dots, u, u+1\}$
$\mathbb{S}_{kt}$	set of dump-to-dump trips for vehicle $k \in \mathcal{K}$ on day $t \in \mathcal{T}$	$\mathcal{S}$	set of containers in a trip in $\mathbb{S}_{kt}$

Parameters	
ξ_{itg}	Poisson rate for deposit volume ν of container i on day t
$\boldsymbol{\kappa}_{it}$	vector of covariates for container i on day t
$\boldsymbol{\gamma}_{\nu}$	vector of estimable parameters for deposit volume ν
ρ_{it}	demand of container i on day t (random variable)
ε_{it}	error term of container i on day t
υ	forecasting model error (standard deviation of the fit's residuals)
π_{ij}	travel distance of arc (i, j)
τ_{ijk}	travel time of vehicle k on arc (i, j)
λ_i, μ_i	lower and upper time window bound at point i
δ_i	service duration at point i
ω_i	capacity of container i
χ	container overflow cost (monetary)
ζ	container emergency collection cost (monetary)
σ_{it}	1 indicates that container i is in a state of full and overflowing on day t, 0 otherwise
φ_k	daily deployment cost of vehicle k (monetary)
β_k	unit-distance running cost of vehicle k (monetary)
θ_k	unit-time running cost of vehicle k (monetary)
α_{kt}	1 if vehicle k is available on day t, 0 otherwise
α_{ik}	1 if container i is accessible by vehicle k, 0 otherwise
Ω_k	capacity of vehicle k
H	maximum tour duration
ψ	Route Failure Cost Multiplier (RFCM) $\in [0, 1]$
$C_{\mathcal{S}}$	the average routing cost of going from $\mathcal{S} \in \mathbb{S}_{kt}$ to the nearest dump and back to $\mathcal{S}$
h_i	inventory holding cost at point i

Decision Variables	
x_{ijkt}	1 if vehicle k traverses arc (i, j) on day t, 0 otherwise (binary)
y_{ikt}	1 if vehicle k visits point i on day t, 0 otherwise (binary)
z_{kt}	1 if vehicle k is used on day t, 0 otherwise (binary)
q_{ikt}	expected pickup quantity by vehicle k from container i on day t (continuous)
Q_{ikt}	expected cumulative quantity on vehicle k at point i on day t (continuous)
I_{it}	expected inventory of container i at the start of day t (continuous)
S_{ikt}	start-of-service time of vehicle k at point i on day t (continuous)

ability:

$$\min_{\Gamma} \sum_{i \in \mathcal{P}} \sum_{t \in \mathcal{H}} \left(\rho_{it}^{o} - \sum_{v \in \mathcal{V}} \nu \xi_{itv} \right)^2, \tag{2}$$

assuming strict exogeneity and with errors represented by white noise:

$$\rho_{it} = \mathbb{E}(\rho_{it}) + \varepsilon_{it}, \quad \text{where } \varepsilon_{it} \text{ are iid normal}, \tag{3}$$

and where a consistent estimate of the variance is given by:

$$\upsilon^2 = \frac{\sum_{i \in \mathcal{P}} \sum_{t \in \mathcal{H}} \left(\rho_{it}^{o} - \mathbb{E}\left(\rho_{it}\right)\right)^2}{|\mathcal{P}||\mathcal{H}| - \#\text{params}}. \tag{4}$$

The denominator in formula (4) is the total number of data observations $|\mathcal{P}||\mathcal{H}|$ minus the number of estimated parameters in the model. For a more detailed description of the model, the reader is referred to Markov et al. (2015).

3.2 Stochastic IRP Model

Our SIRP is defined for a planning horizon $\mathcal{T} = \{0, \dots, u\}$, and for each day t we are given a complete directed graph $\mathcal{G}(\mathcal{N}, \mathcal{A})$, with $\mathcal{N} = \{o\} \cup \{d\} \cup \mathcal{D} \cup \mathcal{P}$, where o and d represent the depot as an origin and a destination, respectively, $\mathcal{D}$ is the set of dumps, $\mathcal{P}$ is the set of containers, and $\mathcal{A} = \{(i, j) : \forall i, j \in \mathcal{N}, i \neq j\}$ is the set of arcs. For modeling purposes, it is assumed that the set $\mathcal{D}$ contains a sufficient number of replications of each dump to allow multiple visits by the same vehicle on the same day.

There is an asymmetric distance matrix, with π_{ij} the travel distance of arc (i, j). Each vehicle may have a different average speed, which results in a vehicle-specific travel time matrix, where τ_{ijk} is the travel time of vehicle k on arc (i, j). Each point has a single time window $[\lambda_i, \mu_i]$, where λ_i and μ_i stand for the earliest and latest possible start-of-service time. Start of service after μ_i is not allowed, and if the vehicle arrives before λ_i, it has to wait. Service duration at each point is denoted by δ_i. For containers it is mostly influenced by the type of container, e.g. underground or overground, and for dumps by factors such as weighing and billing. Hence service duration is not indexed for vehicle. Service duration at the depots is zero. There is an expected demand $\mathbb{E}(\rho_{it})$ for container i on day t. Container capacity is denoted by ω_i, and a cost χ is charged for a full and overflowing container. There is a heterogeneous fixed fleet $\mathcal{K}$, with each vehicle defined by its capacity Ω_k, a daily deployment cost φ_k, a unit-distance running cost β_k, and a unit-time running cost θ_k. The binary flags α_{kt} denote whether vehicle k is available on day t, and the binary flags α_{ik} denote whether container i is accessible by vehicle k. The maximum tour duration is denoted by H.

We introduce the following binary decision variables: $x_{ijkt} = 1$ if vehicle k traverses arc (i, j) on day t, 0 otherwise; $y_{ikt} = 1$ if vehicle k visits point i on day t, 0 otherwise; $z_{kt} = 1$ if vehicle k is used on day t, 0 otherwise. In addition, the following continuous variables are used: q_{ikt}

for the expected pickup quantity by vehicle k from container i on day t; Q_{ikt} for the expected cumulative quantity on vehicle k at point i on day t; I_{it} for the expected inventory of container i at the start of day t; and S_{ikt} for the start-of-service time of vehicle k at point i on day t. The inventory levels at the start of the planning horizon I_{i0} are known with certainty. For modeling purposes, we assume that container inventory is updated at the start of each day before vehicle visits. Thus, the pickup quantity is independent of the time of day that the vehicle collects a container.

3.2.1 Derivation of the Overflow Probabilities.

Unlike in most traditional IRPs, we have no inventory holding costs at the containers or dumps. To formulate the objective function, we introduce the notions of a regular and an emergency collection. Let σ_{it} denote the state of container i on day t, where $\sigma_{it} = 0$ denotes that container i is not full on day t, while $\sigma_{it} = 1$ denotes that it is full and overflowing. A regular collection of container i on day t by vehicle k is one for which $y_{ikt} = 1$. On the other hand, an emergency collection occurs when the container is in a state $\sigma_{it} = 1$ and for $y_{ikt} = 0, \forall k \in \mathcal{K}$. An emergency collection incurs a high cost ζ, which is an approach often employed in the IRP literature (e.g. Dror and Ball, 1987; Trudeau and Dror, 1992; Hemmelmayr et al., 2010; Coelho et al., 2014a), and empties the container in question. Our routing cost is thus counterbalanced by the container overflow cost χ and the emergency collection cost ζ which, due to embedded conditionality, lead to a non-linear objective function. It should be mentioned that there is an important conceptual difference between χ and ζ. While the former has a well-defined monetary value which the collector bears in case of container overflow, the latter is a parameter that needs to be calibrated to represent the average actual cost of emergency collection or to otherwise reflect the collector's policy in such cases.

Assume that we start with an initial inventory I_{i0} such that container i is initially in state $\sigma_{i0} = 0$. If the container never undergoes a regular collection during the planning horizon, its state probability tree develops as illustrated in Figure 2. We observe that all branches starting from a state $\sigma_{it} = 0$ involve the calculation of conditional probabilities, while those starting from a state $\sigma_{it} = 1$ involve unconditional probabilities because the inventory is set to zero by the emergency collection. For our problem, we are only interested in the probability of overflow, i.e. of being in a state $\sigma_{it} = 1$. For day $t = 0$, this is either 0 or 1, depending on the initial state, while for all other days it is obtained by successively multiplying the branch probabilities. If we impose a regular collection on day $t = 2$, the probability of overflow on day $t = 2$ is the probability of being in state $\sigma_{i2} = 1$. To calculate the probability of overflow for subsequent days, we start a new tree with a root on day $t = 2$. Without loss of generality, we can set the root of the new tree to state $\sigma_{i2} = 1$ since the inventory is set to zero by the regular collection and the two initial branches of the new tree will have unconditional probabilities. Regardless of the initial state, all branch probabilities can be precomputed, including those that occur when a new tree is started by a regular collection. For container i, the exhaustive list is given by:

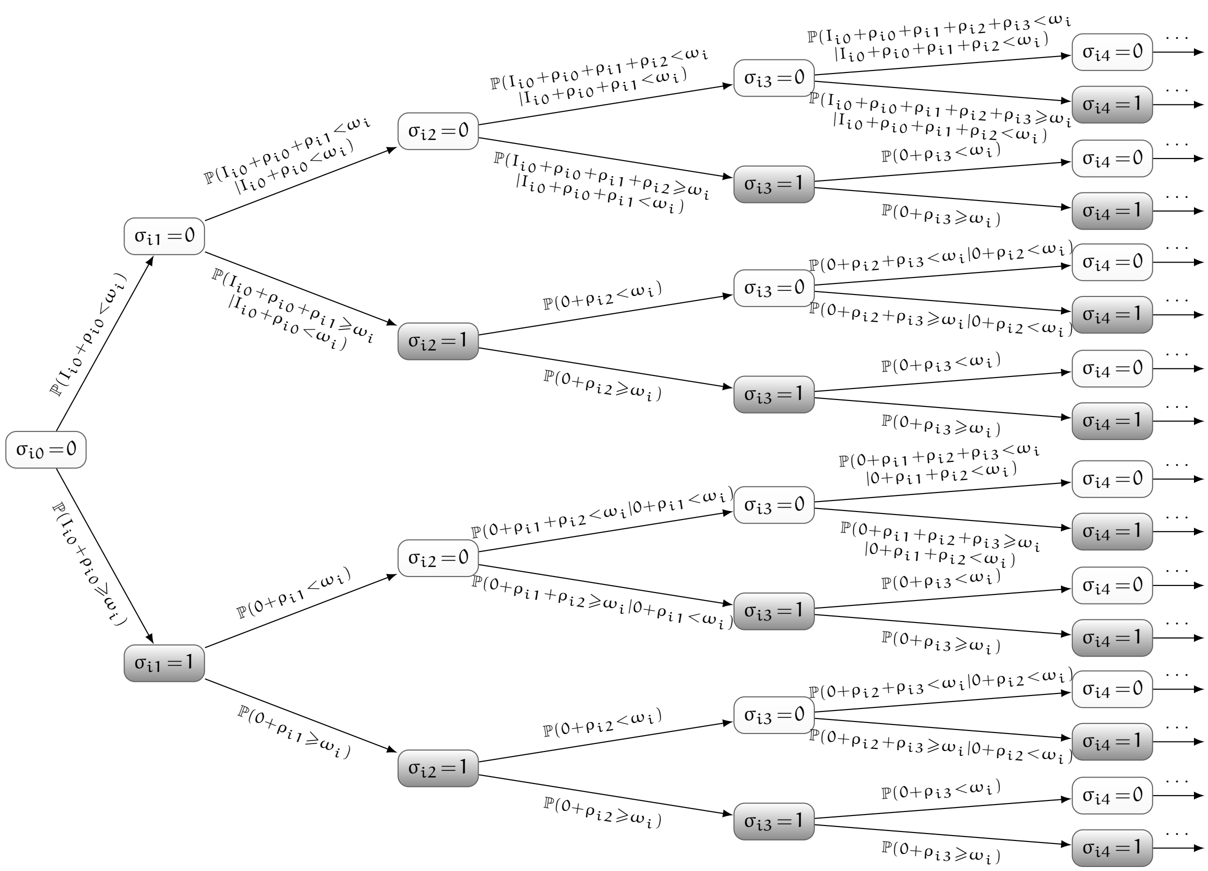
- The unconditional probability of overflow with non-zero initial inventory. This only applies at the root node of the state probability tree on day $t = 0$ and is given by $\mathbb{P}(I_{i0} + \rho_{i0} \geqslant \omega_i)$.

- The unconditional probabilities of overflow with zero initial inventory. These apply either at the root node or at a state of overflow and are expressed by $\mathbb{P}(0 + \rho_{ih} \geqslant \omega_i), \forall h \in \mathcal{T}$.
- The conditional probabilities of overflow with non-zero initial inventory. These apply along the tree's uppermost branch and write as $\mathbb{P}(I_{i0} + \sum_{t=0}^{h} \rho_{it} \geqslant \omega_i \mid I_{i0} + \sum_{t=0}^{h-1} \rho_{it} < \omega_i), \forall h \in \mathcal{T}\colon h > 0$.
- The conditional probabilities of overflow with zero initial inventory apply in all other cases and are obtained as $\mathbb{P}(0 + \sum_{t=g}^{h} \rho_{it} \geqslant \omega_i \mid 0 + \sum_{t=g}^{h-1} \rho_{it} < \omega_i), \forall g, h \in \mathcal{T}\colon h > g$.

The calculation of the conditional probabilities involves the evaluation of:

$$\mathbb{P}\left(I_{ig} + \sum_{t=g}^{h} \rho_{it} \geqslant \omega_i \,\middle|\, I_{ig} + \sum_{t=g}^{h-1} \rho_{it} < \omega_i \right). \tag{5}$$

Figure 2: State Probability Tree Starting from a Non-Full State Without a Regular Collection



Using assumption (3), expression (5) takes the form:

$$\mathbb{P}\left(\sum_{t=g}^{h} \varepsilon_{it} \geqslant \omega_i - I_{ig} - \sum_{t=g}^{h} \mathbb{E}(\rho_{it}) \,\middle|\, \sum_{t=g}^{h-1} \varepsilon_{it} < \omega_i - I_{ig} - \sum_{t=g}^{h-1} \mathbb{E}(\rho_{it})\right). \tag{6}$$

Substitute $a = \omega_i - I_{ig} - \sum_{t=g}^{h-1} \mathbb{E}(\rho_{it})$, and $X = \sum_{t=g}^{h-1} \varepsilon_{it}$, where $X \sim \mathcal{N}(0, (h-g)\upsilon^2)$ and X is independent of ε_{ih}. Formula (6) then rewrites as:

$$\begin{aligned}\mathbb{P}\left(X + \varepsilon_{ih} \geqslant a - \mathbb{E}(\rho_{ih}) \,|\, X < a\right) &= \frac{\mathbb{P}(\varepsilon_{ih} \geqslant a - \mathbb{E}(\rho_{ih}) - X,\, X < a)}{\mathbb{P}(X < a)} = \\ &= \frac{1}{\Phi_X(a)} \times \frac{1}{2\pi\upsilon^2\sqrt{h-g}} \int_{-\infty}^{a} \int_{a-\mathbb{E}(\rho_{ih})-x}^{\infty} e^{-\frac{x^2}{2(h-g)\upsilon^2}} e^{-\frac{y^2}{2\upsilon^2}} \, dx dy,\end{aligned} \tag{7}$$

where $\Phi_X(\cdot)$ is the CDF of X. We standardize the joint probability in expression (7) by setting $x = x/(\upsilon\sqrt{h-g})$ and $y = y/\upsilon$, and thus arrive at expression (8) for the conditional probability we are looking for:

$$\begin{aligned}\mathbb{P}\left(X + \varepsilon_{ih} \geqslant a - \mathbb{E}(\rho_{ih}) \,|\, X < a\right) &= \frac{1}{2\pi\Phi(\frac{a}{\upsilon\sqrt{h-g}})} \int_{-\infty}^{\frac{a}{\upsilon\sqrt{h-g}}} \int_{\frac{a-\mathbb{E}(\rho_{ih})-x\upsilon\sqrt{h-g}}{\upsilon}}^{\infty} e^{-\frac{x^2}{2}} e^{-\frac{y^2}{2}} \, dx dy = \\ &= \frac{1}{2\sqrt{2\pi}\Phi\left(\frac{a}{\upsilon\sqrt{h-g}}\right)} \int_{-\infty}^{\frac{a}{\upsilon\sqrt{h-g}}} e^{-\frac{x^2}{2}} \operatorname{erfc}\left(\frac{a - \mathbb{E}(\rho_{ih}) - x\upsilon\sqrt{h-g}}{\upsilon\sqrt{2}}\right) dx,\end{aligned} \tag{8}$$

where $\Phi(\cdot)$ is the CDF of a standard normal variable. The single integral in expression (8) can be evaluated using a standard statistical package like R in the order of milliseconds. For a problem of realistic size, all the necessary unconditional and conditional probabilities can be automatically precomputed in a negligible amount of time using the latest container information.

3.2.2 Objective Function.

We are now in a position to formulate the objective function z which comprises the Expected Overflow and Emergency Collection Cost (EOECC), the Routing Cost (RC), and the Expected Route Failure Cost (ERFC):

$$\min z = \text{EOECC} + \text{RC} + \text{ERFC}. \tag{9}$$

The expected overflow and emergency collection cost is expressed as:

$$\text{EOECC} = \sum_{t \in \mathcal{T} \cup \mathcal{T}^+} \sum_{i \in \mathcal{P}} \left(\mathbb{P}\left(\sigma_{it}{=}1 \,|\, m{=}\max\left(0, g{<}t\colon \exists k \in \mathcal{K}\colon y_{ikg}{=}1\right)\right) \left(\chi + \zeta - \zeta \sum_{k \in \mathcal{K}} y_{ikt}\right)\right), \tag{10}$$

where the probability of being in a state of overflow is conditional on the most recent regular collection, identified for each container i by the index m. For a given container i, the max operator returns the day g of the most recent regular collection, or 0 if there have been no regular collections before day t. The state probability is calculated by multiplication of the involved branch probabilities on the tree, where conditional probabilities are computed using formula (5). For a day t, the applied cost includes the container overflow cost χ and the emergency collection cost ζ in case there is no regular collection on that day, and only the container overflow cost χ in case there is a regular collection. Although there is no uncertainty on day $t = 0$, we still need to pay the overflow cost if the container is in a state of overflow. On the other hand, the inventories at the start of the first day after the end of the planning horizon are completely determined by the decisions taken during the planning horizon. For this reason, EOECC is computed for $t \in \mathcal{T} \cup \mathcal{T}^+$, where $\mathcal{T}^+ = \{1, \ldots, u, u+1\}$.

The routing cost does not consider the probability of skipping containers because they overflowed earlier than expected and underwent an emergency collection. Trudeau and Dror (1992) develop a probabilistic expression for a similar problem with the assumption of a single visit per customer and a single permitted stock-out during the planning horizon. We do not impose these two assumptions on our problem, and moreover, unlike in Trudeau and Dror (1992), our routing cost includes a time component as well, which means that if a point is removed from or added to the tour, we need to recalculate waiting times and remove any slack. Therefore, we prefer to keep the deterministic expression for the routing cost with the risk of it slightly overestimating the expected routing cost:

$$\mathrm{RC} = \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} \left(\varphi_k z_{kt} + \beta_k \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} \pi_{ij} x_{ijkt} + \theta_k \left(S_{dkt} - S_{okt} \right) \right). \tag{11}$$

The expected route failure cost reflects the vehicles' inability, due to insufficient capacity, to serve the containers on the scheduled depot-to-dump or a dump-to-dump trips. Compared to Trudeau and Dror's (1992) work, our work has the additional complexity of multiple and unlimited dump visits. Thus, similar to the case of the routing cost above, we develop the more tractable yet realistic expression:

$$\mathrm{ERFC} =$$

$$= \sum_{t \in \mathcal{T} \setminus 0} \sum_{k \in \mathcal{K}} \sum_{\mathcal{S} \in \mathcal{S}_{kt}} \left(\psi C_{\mathcal{S}} \, \mathbb{P} \left(\sum_{s \in \mathcal{S}} \left(\Lambda_{sm} + \sum_{h=m}^{t-1} \rho_{sh} \right) > \Omega_k \;\middle|\; m = \max(0, g < t \colon \exists k' \in \mathcal{K} \colon y_{sk'g} = 1) \right) \right), \tag{12}$$

where $\mathcal{S}_{kt}$ is the set of depot-to-dump or dump-to-dump trips for vehicle k on day t, $\mathcal{S}$ is the set of containers in a particular trip, $C_{\mathcal{S}}$ is the average routing cost of going from this set to the nearest dump and back, and Λ_{sm} is the inventory of container s after regular collection on day m. The parameter $\psi \in [0, 1]$, which we refer to as the Route Failure Cost Multiplier (RFCM), is used to scale up or down the degree of conservatism regarding this cost component. The probability is conditional on the most recent regular collection identified for each container s by the index m. For a given container s, the max operator returns the day g of the most recent regular collection, or 0 if the container has not undergone any regular collections before day t. The inventory of container s after regular collection on day m is defined as:

$$\Lambda_{sm} = I_{sm} - \sum_{k \in \mathcal{K}} q_{skm} \, . \tag{13}$$

Given an OU inventory policy, this value is 0 if there is a regular collection on day m, and is equal to the initial inventory I_{s0} if there is no regular collection on day 0. Thus, the probability of a route failure in a set $\mathcal{S}$ is the probability that the sum of the random daily demands, plus potentially the initial inventories on day 0, collected from this set exceeds the vehicle capacity. By definition, there are no route failures on day $t = 0$ as the container information is fully known. We consider it safe to assume only one route failure per depot-to-dump or dump-to-dump trip. It would indeed be very improbable that the expected inventories were so far from the realized ones that they were more than twice as high for a given trip. Again, here we do not consider the probability of skipping containers because they underwent an emergency collection earlier. Hence, like the expected routing cost, the expected route failure cost is a slight overestimation.

The nearest dump to each container can be precomputed. Probability-wise, once the days g of the previous collection of each container are found, the remaining probability is unconditional. Given that it involves multiple containers, it is impractical to precompute for all combinations. Thus, we implement a solution in which the probability is evaluated during runtime using an approximation of the standard normal distribution based on the approximation of the error function:

$$\text{erf}(x) \approx 1 - \left(a_1 t + a_2 t^2 + \cdots + a_5 t^5\right) e^{-x^2},\ t = \frac{1}{1 + dx}, \tag{14}$$

where $d = 0.3275911$, $a_1 = 0.254829592$, $a_2 = -0.284496736$, $a_3 = 1.421413741$, $a_4 = -1.453152027$, $a_5 = 1.061405429$, and whose maximum approximation error is 1.5×10^{-7} (Abramowitz and Stegun, 1972). These repetitive calculations have no discernible influence on the algorithm's runtime.

3.2.3 Constraints.

The constraints are extended from the VRP with intermediate facilities presented in Markov et al. (2016) and can be split into several categories, with the first category consisting of basic vehicle routing constraints. Equalities (15) and (16) ensure that only available vehicles are used, and that if a vehicle is used, its tour starts at the origin and ends at the destination, with a visit to a dump immediately before that. Constraints (17) link the visit and the routing variables, while constraints (18) stipulate that a container is visited by at most one vehicle on a given day. Inequalities (19) guarantee that vehicles do not visit inaccessible points. Flow conservation is represented by constraints (20).

$$\sum_{j \in \mathcal{N}} x_{ojkt} = \alpha_{kt} z_{kt}, \qquad \forall t \in \mathcal{T}, k \in \mathcal{K} \tag{15}$$

$$\sum_{i \in \mathcal{D}} x_{idkt} = \alpha_{kt} z_{kt}, \qquad \forall t \in \mathcal{T}, k \in \mathcal{K} \tag{16}$$

$$y_{ikt} = \sum_{j \in \mathcal{N}} x_{ijkt} = \sum_{j \in \mathcal{N}} x_{jikt}, \qquad \forall t \in \mathcal{T}, k \in \mathcal{K}, i \in \mathcal{P} \tag{17}$$

$$\sum_{k\in\mathcal{K}} y_{ikt} \leqslant 1, \qquad \forall t \in \mathcal{T}, i \in \mathcal{P} \tag{18}$$

$$y_{ikt} \leqslant \alpha_{ik}, \qquad \forall t \in \mathcal{T}, k \in \mathcal{K}, i \in \mathcal{P} \tag{19}$$

$$\sum_{i\in\mathcal{N}} x_{ijkt} = \sum_{i\in\mathcal{N}} x_{jikt}, \qquad \forall t \in \mathcal{T}, k \in \mathcal{K}, j \in \mathcal{D} \cup \mathcal{P} \tag{20}$$

The inventory constraints are necessary for tracking the container inventories and linking them to the vehicle visits and the pickup quantities. Equalities (21) track the inventories as a function of the previous day's inventories, pickup quantities and expected demands. Constraints (22) impose the fact that, in expected terms, we do not accept container overflows. As already mentioned in Section 3.2.1, the inventories need to be computed over $\mathcal{T}^+$, starting from the fully known inventories on day $t = 0$. Constraints (23) ensure that if the starting inventory exceeds capacity, the container must be collected on day $t = 0$. The big-M reflects the assumption that the expected daily demand can never exceed the container capacity. In addition, a daily rolling horizon enforces the one-day back-order limit. Inequalities (24) force the pickup quantity to zero if the container is not visited. Inequalities (25) and (26) represent the OU policy. The big-M values in constraints (24) and (26) can be set to $2\omega_i$ for $t = 0$ and to ω_i otherwise, reflecting the fact that the picked-up inventory can never exceed capacity, except on day $t = 0$.

$$I_{it} = I_{i(t-1)} - \sum_{k\in\mathcal{K}} q_{ik(t-1)} + \mathbb{E}(\rho_{i(t-1)}), \qquad \forall t \in \mathcal{T}^+, i \in \mathcal{P} \tag{21}$$

$$I_{it} \leqslant \omega_i, \qquad \forall t \in \mathcal{T}^+, i \in \mathcal{P} \tag{22}$$

$$I_{i0} - \omega_i \leqslant \omega_i \sum_{k\in\mathcal{K}} y_{ik0}, \qquad \forall i \in \mathcal{P} \tag{23}$$

$$q_{ikt} \leqslant M y_{ikt}, \qquad \forall t \in \mathcal{T}, k \in \mathcal{K}, i \in \mathcal{P} \tag{24}$$

$$q_{ikt} \leqslant I_{it}, \qquad \forall t \in \mathcal{T}, k \in \mathcal{K}, i \in \mathcal{P} \tag{25}$$

$$q_{ikt} \geqslant I_{it} - M(1 - y_{ikt}), \qquad \forall t \in \mathcal{T}, k \in \mathcal{K}, i \in \mathcal{P} \tag{26}$$

In the context of vehicle capacities, inequalities (27) limit the cumulative quantity on the vehicle at each container, while equalities (28) reset it to zero at the origin, destination, and dumps. Keeping track of the cumulative quantity on the vehicle is achieved by constraints (29).

$$q_{ikt} \leqslant Q_{ikt} \leqslant \Omega_k, \qquad \forall t \in \mathcal{T}, k \in \mathcal{K}, i \in \mathcal{P} \tag{27}$$

$$Q_{ikt} = 0, \qquad \forall t \in \mathcal{T}, k \in \mathcal{K}, i \in \mathcal{N} \setminus \mathcal{P} \tag{28}$$

$$Q_{ikt} + q_{jkt} \leqslant Q_{jkt} + \Omega_k \left(1 - x_{ijkt}\right), \qquad \forall t \in \mathcal{T}, k \in \mathcal{K}, i \in \mathcal{N} \setminus \{d\}, j \in \mathcal{P} \tag{29}$$

The next four constraints express the intra-day temporal characteristics of the problem. Inequalities (30) calculate the start-of-service time at each point. In addition, these constraints eliminate the possibility of subtours and ensure that a point is not visited more than once by the same vehicle. Constraints (31), (32) and (33) enforce the time windows and maximum tour duration.

$$S_{ikt} + \delta_i + \tau_{ijk} \leqslant S_{jkt} + \left(\mu_i + \delta_i + \tau_{ijk}\right)\left(1 - x_{ijkt}\right), \ \forall t \in \mathcal{T}, k \in \mathcal{K}, i \in \mathcal{N} \setminus \{d\}, j \in \mathcal{N} \setminus \{o\} \tag{30}$$

$$\lambda_i \sum_{j \in \mathcal{N}} x_{ijkt} \leqslant S_{ikt}, \qquad \forall t \in \mathcal{T}, k \in \mathcal{K}, i \in \mathcal{N} \setminus \{d\} \tag{31}$$

$$S_{jkt} \leqslant \mu_j \sum_{i \in \mathcal{N}} x_{ijkt}, \qquad \forall t \in \mathcal{T}, k \in \mathcal{K}, j \in \mathcal{N} \setminus \{o\} \tag{32}$$

$$0 \leqslant S_{dkt} - S_{okt} \leqslant H \qquad \forall t \in \mathcal{T}, k \in \mathcal{K} \tag{33}$$

Finally, lines (34) and (35) establish the variable domains.

$$x_{ijkt}, y_{ikt}, z_{kt} \in \{0, 1\}, \qquad \forall t \in \mathcal{T}, k \in \mathcal{K}, i, j \in \mathcal{N} \tag{34}$$

$$q_{ikt}, Q_{ikt}, I_{it}, S_{ikt} \geqslant 0, \qquad \forall t \in \mathcal{T}, k \in \mathcal{K}, i \in \mathcal{N} \tag{35}$$

3.3 Model Reformulations for Solving Benchmark Instances

The model presented above describes a problem that is not encountered in the literature. In order to evaluate the performance of the solution methodology presented in Section 4, we benchmark it to less general problems from the literature. Below, we present the reformulations of the original model to solve selected IRP and VRP instances from the literature. The corresponding modifications to the general solution methodology are described in Section 4.3.

3.3.1 IRP Reformulation.

For the benchmark IRP instances from the literature, we assume a distribution context and the presence of inventory holding costs at the depot and the customers. There are no intermediate facilities. We are in a deterministic setting and $\rho_{it} = \mathbb{E}(\rho_{it})$. The commodity becomes available at the depot at a rate ρ_{ot} on day t and is consumed by customer i at a rate ρ_{it} on day t. Let h_o and h_i denote the inventory holding cost per day at the depot and customer i, respectively. In addition, we redefine q_{ikt} as the quantity delivered by vehicle k to customer i on day t, and Q_{ikt} as the cumulative quantity delivered by vehicle k when arriving at point i on day t. The objective function of the benchmark IRP is composed of the inventory holding costs at the depot and the customers, and the routing cost, and writes as:

$$\begin{aligned} \min z^{\mathrm{IRPB}} = & \sum_{t \in \mathcal{T} \cup \mathcal{T}^+} h_o I_{ot} + \sum_{t \in \mathcal{T} \cup \mathcal{T}^+} \sum_{i \in \mathcal{P}} h_i I_{it} \\ & + \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} \left(\varphi_k z_{kt} + \beta_k \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} \pi_{ij} x_{ijkt} + \theta_k \left(S_{dkt} - S_{okt} \right) \right). \end{aligned} \tag{36}$$

A special set of constraints is needed for the inventory definition at the depot. Constraints (37) define the inventory level at the depot as the sum of the previous day's inventory and quantity made available minus the previous day's amount delivered to customers. Inequalities (38) forbid a stock-out at the depot given the total quantity delivered to customers.

$$I_{ot} = I_{o(t-1)} + \rho_{o(t-1)} - \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{P}} q_{ik(t-1)}, \qquad \forall t \in \mathcal{T}^+ \tag{37}$$

$$I_{ot} \geqslant \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{P}} q_{ikt}, \qquad \forall t \in \mathcal{T} \tag{38}$$

We redefine the evolution of the inventory level at the customers for a distribution context. Constraints (39) define the inventory level at the customers as the sum of the previous day's inventory and quantity delivered minus the previous day's demand. Constraints (38) forbid the occurrence of stock-outs.

$$I_{it} = I_{i(t-1)} + \sum_{k\in\mathcal{K}} q_{ik(t-1)} - \rho_{i(t-1)}, \qquad \forall i \in \mathcal{P}, t \in \mathcal{T}^+ \tag{39}$$

$$I_{it} \geqslant 0, \qquad \forall i \in \mathcal{P}, t \in \mathcal{T}^+ \tag{40}$$

The OU policy also needs to be redefined for a distribution context. Constraints (41), (42), and (43) express the fact that if a customer is visited, its inventory is brought up to its capacity.

$$q_{ikt} \geqslant \omega_i y_{ikt} - I_{it}, \qquad \forall t \in \mathcal{T}, k \in \mathcal{K}, i \in \mathcal{P} \tag{41}$$

$$q_{ikt} \leqslant \omega_i - I_{it}, \qquad \forall t \in \mathcal{T}, k \in \mathcal{K}, i \in \mathcal{P} \tag{42}$$

$$q_{ikt} \leqslant \omega_i y_{ikt}, \qquad \forall t \in \mathcal{T}, k \in \mathcal{K}, i \in \mathcal{P} \tag{43}$$

To avoid unnecessary complications in the reformulation, we can safely assume that there is a single dummy dump with zero service time and distance to the depot. The basic routing constraints (15)–(20), the vehicle capacity constraints (27)–(29), the intra-day temporal constraints (30)–(33), and the domain constraints (34)–(35) can thus be reused.

3.3.2 VRP Reformulation.

For the VRP, it suffices to collapse the planning horizon to $\mathcal{T} = \{0\}$ and redefine the objective function z in terms of the Routing Cost (RC) only:

$$\min z^{\text{VRPB}} = \sum_{k\in\mathcal{K}} \left(\varphi_k z_{k0} + \beta_k \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} \pi_{ij} x_{ijk0} + \theta_k \left(S_{dk0} - S_{ok0}\right) \right). \tag{44}$$

For day $t = 0$, demands are deterministic. As far as the constraints are concerned, the inequality sign in constraints (18) of the original model becomes an equality sign, providing that each container is visited by exactly one vehicle. Constraints (21) and (22) are dropped since the VRP is solved for a single period and we disregard its effect on the future. Constraints (23) are dropped as they become redundant given the modified constraints (18). Since there is no inventory tracking over the planning horizon, it is irrelevant whether we are in a collection or in a distribution context.

4 Adaptive Large Neighborhood Search

Adaptive Large Neighborhood Search (ALNS) was introduced by Ropke and Pisinger (2006a) in the context of the pickup and delivery problem with time windows. It is a type of large neighborhood search in which a number of fairly simple operators compete in modifying the current solution. At each iteration of the search process, a number of customers is removed from the current solution by a destroy operator, after which they are reinserted elsewhere by a repair operator. In the context of our IRP, not all customers need to be visited every day, or even at all. Hence, we do not require that all removed customers should be reinserted by the repair operator. The search guiding principle can be based on any metaheuristic framework. Simulated annealing appears to be the preferred approach in the ALNS literature, and is also the one we implement. Given an incumbent solution s, a randomly drawn neighbor solution s' is always accepted if $f(s') < f(s)$, and with probability $\exp(-(f(s') - f(s))/T)$ otherwise, with $f(s)$ representing the solution cost and $T > 0$ the current temperature. The temperature is initialized as T^{start} and is reduced at each iteration by a cooling rate $r \in (0, 1)$. The search stops when T reaches a predetermined T^{end}.

Operator choice is governed by a roulette-wheel mechanism. Each operator i has a weight W_i, which depends on its past performance and a score. Given the set of destroy (repair) operators $\mathcal{O}$, the destroy (repair) operator i is selected with probability $W_i / \sum_{j \in \mathcal{O}} W_j$. The ALNS starts with all weights set to one and all scores set to zero. The scores of the selected destroy-repair couple are increased by e_1 if they find a new best feasible solution, by $e_2 < e_1$ if they improve the incumbent, and by $e_3 < e_2$ if they do not improve the incumbent but the new solution is accepted. This strategy rewards successful operator couples, while at the same time maintaining diversification during the search. It is important to note that if a destroy-repair couple leads to a visited solution, no reward is applied. The search is divided into segments of F iterations each, at the end of which the operator weights are updated. Let C_i^F denote the score of operator i and N_i^F the number of times it was applied in the last segment of length F. The new weights are computed as follows:

$$W_i = \begin{cases} W_i & \text{if } N_i^F = 0, \\ (1-b)W_i + bC_i^F / \left(m_i N_i^F\right) & \text{otherwise.} \end{cases} \quad (45)$$

In expression (45), m_i is a normalization factor damping the weights of more computationally expensive operators by multiplying the number of times they were applied (Ropke and Pisinger, 2006b; Coelho et al., 2012a). The value $b \in [0, 1]$ is a reaction factor, controlling the relative effect of past performance and the scores on the new weights. Once the weights are updated, C_i^F and N_i^F are reset to zero. Algorithm 1 is a pseudocode of the ALNS implementation with simulated annealing. The function $f(\cdot)$ represents the full solution cost including penalties for feasibility violations, as explained in Section 4.1 next. Regarding the initial solution s^{init}, we build empty tours consisting of the depot as an origin and destination and one dump in between, without inserting any containers. An empty tour is built for each available vehicle on each day of the planning horizon. Since the destroy operators will have no effect in the beginning, the repair operators will insert containers and construct a non-empty solution.

Algorithm 1 ALNS Pseudocode

Input initial solution s^{init}
Output best found solution s^{best}

```
all weights equal to 1, all scores equal to 0
s^best ← s ← s^init
T ← T^start
while T ⩾ T^end do
    s' ← s
    select a destroy-repair couple using roulette wheel and apply to s'
    if f(s') < f(s) then
        s ← s'
        if f(s') < f(s^best) and s' is feasible then
            s^best ← s'
            update scores of destroy-repair couple by e_1
        else
            update scores of destroy-repair couple by e_2
        end if
    else if s' is accepted
        s ← s'
        update scores of destroy-repair couple by e_3
    end if
    if iteration count is multiple of F then
        update weights and reset scores to 0
    end if
    T ← (1 − r)T
end while
```

4.1 Solution Representation

To facilitate the search and avoid becoming trapped in local optima, we admit infeasible intermediate solutions at a penalty. This relaxation technique is especially useful for tightly constrained problems. Let s be a solution and let $\mathcal{N}'_{kt}(s)$ denote all point visits by vehicle k on day t in s, where each visit is a replication of the visited point. In addition, let $\mathcal{P}'_{kt}(s) \subset \mathcal{N}'_{kt}(s)$ denote all point visits where the next visit is a dump. We also define the function $(x)^+ = \max\{0, x\}$. Our ALNS admits the following types of intermediate feasibility violations:

1. Vehicle capacity violation is the sum of excess cumulative demand in $\mathcal{P}'_{kt}(s), \forall t \in \mathcal{T}, k \in \mathcal{K}$. Formally, it is defined as:

$$V^{\Omega}(s) = \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{P}'_{kt}} (Q_{ikt} - \Omega_k)^+. \tag{46}$$

2. Time window violation is the total violation of the upper time window bounds μ_i in $\mathcal{N}'_{kt}(s), \forall t \in \mathcal{T}, k \in \mathcal{K}$. Lower time window bounds cannot be violated because if the

vehicle arrives at point i before λ_i, it waits. Hence, formally, we have:

$$V^{\mu}(s) = \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{N}'_{kt}} (S_{ikt} - \mu_i)^+. \tag{47}$$

3. Duration violation is expressed as the sum of excess durations. It is verified after time window violation. For each tour that has no time window violation, we apply forward time slack reduction (Savelsbergh, 1992), which may minimize tour duration while preserving time window feasibility. In mathematical terms, duration violation writes as:

$$V^{H}(s) = \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} (S_{dkt} - S_{okt} - H)^+. \tag{48}$$

4. Container capacity violation is the sum of excess container inventories $\forall t \in \mathcal{T}^+, i \in \mathcal{P}$, or:

$$V^{\omega}(s) = \sum_{t \in \mathcal{T}^+} \sum_{i \in \mathcal{P}} (I_{it} - \omega_i)^+. \tag{49}$$

5. Backorder limit violation is the sum of excess container inventories on day $t = 0, \forall i \in \mathcal{P}$ that are not visited on day $t = 0$. In mathematical terms, this is expressed as:

$$V^{0}(s) = \sum_{i \in \mathcal{P}} \left(\left(1 - \sum_{k \in \mathcal{K}} y_{ik0} \right) (I_{i0} - \omega_i)^+ \right). \tag{50}$$

6. Accessibility violation is the sum of inaccessible visits in $\mathcal{N}'_{kt}(s), \forall t \in \mathcal{T}, k \in \mathcal{K}$. They are accounted for as:

$$V^{\alpha}(s) = \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{N}'_{kt}} (y_{ikt} - \alpha_{ik})^+. \tag{51}$$

Including the possibility of all violations, the complete solution cost during the search is represented by:

$$f(s) = z(s) + L^{\Omega}V^{\Omega}(s) + L^{\mu}V^{\mu} + L^{H}V^{H}(s) + L^{\omega}V^{\omega}(s) + L^{0}V^{0}(s) + L^{\alpha}V^{\alpha}(s). \tag{52}$$

The parameters L^{Ω} through L^{α} are the penalties for each type of feasibility violation. They are dynamically adjusted during the search so as to encourage the exploration of infeasible solutions but to avoid staying infeasible for too long. At each accepted solution, the incumbent s is checked for each type of violation. If it is non-zero, its respective penalty is increased by a factor $\ell \in (0, 1]$, otherwise it is reduced by the same factor. If s has no feasibility violation, the values of $f(s)$ and $z(s)$ coincide. As indicated in Algorithm 1, the update of the best solution requires feasibility with respect to conditions (46) through (51).

4.2 Operators

The main ingredient of the ALNS are the destroy and repair operators. Some of the operators are borrowed or inspired from Coelho et al.'s (2012a) ALNS for the consistent multi-vehicle IRP and from Buhrkal et al.'s (2012) ALNS and Hemmelmayr et al.'s (2013) VNS for the VRP with intermediate facilities. Our particular implementation also accounts for the fact that we have a heterogeneous fixed fleet. We use the following destroy operators:

1. *Remove ν containers randomly.* This operator selects a random tour and removes a random container from it. It is applied ν times, where ν is an integer drawn from a discrete semi-triangular distribution bounded below by 1 and above by the number of containers in $\mathcal{P}$. Small ν's result in cosmetic changes to the solution, while big ν's, which are drawn with a lower probability, lead to larger perturbations.

2. *Remove ν worst containers.* This operator removes the container that would lead to the largest savings in the solution cost. It is applied ν times.

3. *Shaw removals.* Based on the ideas of Shaw (1997) and Ropke and Pisinger (2006a), this operator removes customer clusters. In particular, it selects a random tour and a random container i in it. It then proceeds to find the distance π_{ij}^{min} to the closest container j in the same tour, and removes container i as well as all containers from this tour that are within $2\pi_{ij}^{min}$ of i. If the selected container i is the only container in the tour, it is the only one that is removed.

4. *Empty a random day.* This operator selects a random day and empties all tours performed on it.

5. *Empty a random vehicle.* This operator selects a random vehicle and empties the tours performed by it on all days.

6. *Remove a random dump.* This operator selects a random tour and a random dump in it, excluding the last dump, and removes it.

7. *Remove the worst dump.* This operator removes the dump that would lead to the largest savings in the solution cost. The last dump in each tour is never removed.

8. *Remove consecutive visits.* This operator inspects each container over the planning horizon and, if it is visited on two consecutive days, removes the second visit. This is based on the idea that optimal or good-quality solutions will rarely have container visits on consecutive days.

In addition, we use the following repair operators:

1. *Insert ν containers randomly.* This operator selects a random tour and a random container from $\mathcal{P}$ not visited on the day the tour is performed, and inserts it in the tour using best insertion, i.e. in the position in the selected tour that would lead to the smallest increase in the solution cost. It is applied ν times.

2. *Insert ν containers in the best way.* This operator selects a random container from $\mathcal{P}$ and inserts it in the tour and in the position that would lead to the smallest increase in the solution cost, checking that the container is not visited on the day the tour is performed. It is applied ν times.

3. *Shaw insertions.* This operator selects a random tour and a random container $i \in \mathcal{P}$ not visited on the day the tour is performed. It then proceeds to find the distance $\pi_{ij}^{\min}$ to the closest container $j \in \mathcal{P}$ also not visited on this day. It inserts into the selected tour the container i as well as all containers not served on this day that are within $2\pi_{ij}^{\min}$ of i using best insertion.

4. *Swap ν random containers.* This operator selects two random tours and a random container in each one, and swaps the container-to-tour assignment by using best insertion in each tour. It is applied ν times.

5. *Insert a dump randomly.* This operator selects a random tour and a random dump from $\mathcal{D}$ and inserts it at a random position in the tour.

6. *Insert a dump in the best way.* This operator selects a random dump from $\mathcal{D}$ and inserts it in the tour and in the position that would lead to the smallest increase in the solution cost.

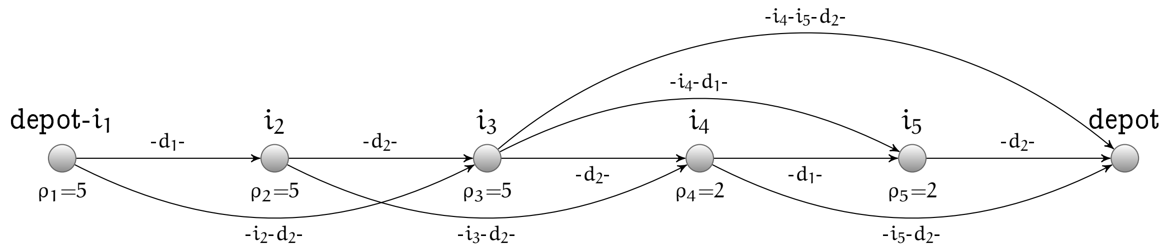
7. *Swap random dumps.* This operator selects two random tours and a random dump in each one, and swaps the dumps.

8. *Replace a random dump.* This operator selects a random tour and a random dump in it, and replaces it with another random dump from $\mathcal{D}$.

9. *Reorder dumps.* Based on the idea of Hemmelmayr et al. (2013), this operator selects a random tour, removes all dump visits from it, and finds the locally optimal dump visit configuration that preserves vehicle capacity feasibility. Figure 3 provides an illustrative example of a tour starting at the depot, visiting containers i_1 through i_5, and terminating at the depot. The values of ρ_1 through ρ_5 denote the container demands, and we assume a vehicle with a capacity of 10 units. Because a dump will never be visited between the depot and the first container, they can be merged into a single node. Each arc starts at a container and ends at a container or the depot, visiting on its way the indicated containers and the best dump, either d_1 or d_2, before the end node. The resulting directed graph is not necessarily complete, as it only contains the vehicle capacity preserving arcs. The solution to the problem amounts to finding the shortest path from the origin to the destination node representing the depot. We use the Bellman-Ford algorithm and post-optimize the result using 2-opt local search.

The destroy operators that empty a random day and a random vehicle leave the affected tour with the depot as an origin and destination, and a dump, and the cost of such a tour is considered zero. Thus, all original tours always remain available during the search for removal of points from or insertion of points into. This is a straightforward way to manage the presence of a heterogeneous fixed fleet without having to re-evaluate periodically vehicle-to-

Figure 3: Feasibility Graph of the Reorder Dumps Operator



tour assignments. This strategy will likely not be applicable to more classical metaheuristics that exploit much smaller neighborhoods.

4.3 Algorithmic Modifications for Solving Benchmark Instances

Several modifications to the original ALNS algorithm are needed in order to integrate the model reformulations described in Section 3.3 and necessary for solving the benchmark IRP and VRP instances.

4.3.1 Modifications for the Benchmark IRP.

The benchmark IRP considers no intermediate facilities and so we disregard all dump-related operators. To avoid further changes to the algorithm, the always-present dump visit before the destination depot is created as a dummy node with zero service time and distance to the depot, and as such does not affect the solution. For a given solution s, we define the depot inventory violation as expressed by constraints (38) in the benchmark IRP reformulation:

$$V^I(s) = \sum_{t \in \mathcal{T}} \left(\sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{P}} q_{ikt} - I_{ot} \right)^+. \tag{53}$$

The violation $V^I(s)$ is multiplied by parameter L^I and added to the objective function representation $f(s)$ as in expression (52). Additionally, we redefine the container capacity violation (49) in terms of stock-out as it applies to a distribution context:

$$V^\omega(s) = \sum_{t \in \mathcal{T}^+} \sum_{i \in \mathcal{P}} (-I_{it})^+. \tag{54}$$

The back-order violation (50) is dropped since back-orders do not apply to the benchmark IRP.

4.3.2 Modifications for the Benchmark VRP.

In the original ALNS algorithm, the number of containers inserted into the solution by a repair operator is randomly drawn and not necessarily the same as the number of containers removed by the destroy operator applied before it. This design allows to vary the number of containers visited each day, as this is a decision variable in the IRP. Contrarily, the VRP assumes that all containers are visited in the solution. To achieve the latter, we implement an initial solution construction procedure and a simple rearrangement of the destroy and repair operators.

To construct an initial solution, we use repair operator number 1, *insert* ν *containers randomly*, to insert all containers into the solution. The resulting initial solution is not necessarily feasible. Then we redefine the operators so that all destroy operators and repair operators 4 through 9 are now drawn first, while repair operators 1 through 3 are drawn second. This separation is based on the operators' ability to reinsert containers into the solution. In other words, the repair operators are now only those that have this ability, namely *insert* ν *containers randomly*, *insert* ν *containers in the best way*, and *Shaw insertions*. Moreover, the number of containers to be reinserted is not random. The repair operators now reinsert all containers that were previously removed by the destroy operator. If the destroy operator did not remove any containers, the repair operator is not applied. Given that all containers are now visited in the solution, we drop violations (49) and (50) from the solution representation, i.e. container capacity violation and backorder violation. If the problem at hand considers no intermediate facilities, we disregard all dump-related operators. The always-present dump visit before the destination depot in this latter case is created as a dummy node with zero service time and distance to the depot, and as such does not affect the solution.

5 Numerical Experiments

The ALNS is implemented as a single-thread application in Java, and the forecasting model and the probability calculator for the state probability tree (Figure 2) are scripted in R. All tests have been carried out on a 3.33 GHz Intel Xeon X5680 server running a 64-bit Ubuntu 12.04.5. Each instance is solved 10 times, out of which we report the best and average result, unless indicated otherwise. Section 5.1 describes how the algorithmic parameters were tuned. This is followed by results of the ALNS performance on benchmark IRP and VRP instances in Section 5.2. Finally, Section 5.3 presents an extensive analysis of the model and solution methodology applied on instances derived from real data.

5.1 Parameter Tuning

All algorithmic parameters were tuned on the Archetti et al. (2007) instances for which optimal solutions are available. We first tuned the SA-related parameters followed by the ALNS-related parameters. Initial values were either borrowed from ALNS implementations in the literature or based on preliminary trial-and-error combinations. The parameters were tuned one by one, unless indicated otherwise, in the order in which they appear in Table 3. The initial temperature was set sufficiently high for an initial feasible solution to be found without difficulty. Once this is the case, the temperature is calibrated so that the probability of accepting a solution which is worse than it by a factor of w is 50%. The purpose of this

Table 3: Algorithmic Parameters

SA-Related		ALNS-Related	
Parameter	Value	Parameter	Value
Initial temperature (T^{start})	10,000	F segment length	2000
Start temperature control parameter (w)	0.6	Reaction factor (b)	0.5
Cooling rate (r)	0.99998	Reward e_1	30
Final temperature (T^{end})	0.01	Reward e_2	20
Penalty change rate (ℓ)	1.06	Reward e_3	5

strategy is to limit the search at very high temperatures (Ropke and Pisinger, 2006a). The cooling rate typically results in several hundred thousand iterations on the Archetti et al. (2007) instances, and the final temperature allows sufficient time for the algorithm to converge. The penalty change rate multiplies or divides the penalties associated with conditions (46) through (51) as explained in Section 4.1. After fixing the SA-related parameters, we tuned the ALNS-related parameters. The rewards were tuned together, and after testing several configurations we chose one that attributes a relatively lower reward e_3 for a non-improving but accepted solution. Two of the operators, namely *remove* ν *worst containers* and *insert* ν *containers in the best way*, were given a normalization factor m_i of 4.5. The normalization factors for the rest are all equal to one. All numerical experiments are performed with this parameter setting, unless indicated otherwise.

5.2 Benchmark Results

The model reformulations and algorithmic modifications presented in Section 3.3 and Section 4.3, respectively, were necessary in order to assess the performance of the solution methodology on benchmark instances from the literature. Section 5.2.1 and Section 5.2.2 below present the results obtained by the ALNS on classical IRP and VRP instance testbeds.

5.2.1 Results on IRP Benchmarks.

We ran the ALNS algorithm on the IRP benchmark set proposed by Archetti et al. (2007), which is the first classical IRP testbed in the literature. It represents a deterministic IRP in a distribution context where an inventory holding cost h_o applies at the depot, and inventory holding costs h_i apply at the customers. There is a single vehicle available each day, with its daily deployment cost φ_k and unit-time running cost θ_k both equal to zero, and its unit-distance running cost $\beta_k = 1$. Stock-outs are forbidden at the customers and the depot. Vehicle tours are only limited by the vehicle capacity, and no rich VRP features such as intermediate facilities, time windows, or a maximum tour duration are considered.

The set includes two equal subsets with high, respectively low, inventory holding costs h_i.

The length of the planning horizon $|\mathcal{T}|$ is either 3 or 6 periods, and the number of customers n varies from 5 to 50 for $|\mathcal{T}| = 3$, and from 5 to 30 for $|\mathcal{T}| = 6$. Five instances are generated for each combination of h_i, $|\mathcal{T}|$ and n, thus resulting in a total of 160 instances. Using a branch-and-cut algorithm, Archetti et al. (2007) solve with a proof of optimality all instances except one (low h_i, $|\mathcal{T}| = 3$, $n = 50$), where the gap is brought to 0.99% within the time limit of two hours. A number of heuristic algorithms are tested on these instances or derivations thereof (Archetti et al., 2012; Coelho et al., 2012a,b). The most successful one is the hybrid heuristic of Archetti et al. (2012) which is able to achieve an optimality gap of 0.1% for the OU policy based on a single experiment per instance, and with computation times up to several thousand seconds for the largest instances on an Intel Dual Core 1.86 GHz processor.

Tables 4 and 5 report our results on the Archetti et al. (2007) instances with high and low inventory holding costs, respectively. Each row represents the average over the five instances for each combination of h_i, $|\mathcal{T}|$ and n. The first two columns report the number of periods $|\mathcal{T}|$ in the planning horizon and the number of customers n. This is followed by two versions of the ALNS. The one labeled "Fast" runs on the parameter configuration from Table 3, while the one labeled "Slow" uses a cooling rate r of 0.999995 resulting in more iterations. For each of them, we report the computation time in seconds, and the percent gap of the best and average value of the objective function over 10 runs. Our results are comparable to the best from the literature. The slow version attains a best gap of 0.02% and 0.06%, and an average gap of 0.11% and 0.30% respectively, on the high and low inventory holding cost instances. In

Table 4: Results on Archetti et al. (2007) Instances with High Inventory Holding Cost

		ALNS Fast Version			ALNS Slow Version		
$\|\mathcal{T}\|$	n	Runtime(s.)	Best Gap(%)	Avg Gap(%)	Runtime(s.)	Best Gap(%)	Avg Gap(%)
3	5	8	0.00	0.00	32	0.00	0.00
3	10	14	0.00	0.00	59	0.00	0.00
3	15	22	0.00	0.00	93	0.00	0.00
3	20	36	0.00	0.01	149	0.00	0.00
3	25	53	0.00	0.06	221	0.00	0.01
3	30	77	0.00	0.27	318	0.00	0.06
3	35	108	0.01	0.15	440	0.00	0.04
3	40	149	0.12	0.48	602	0.01	0.23
3	45	199	0.17	0.47	808	0.10	0.25
3	50	276	0.15	0.52	1074	0.07	0.25
6	5	14	0.00	0.00	55	0.00	0.00
6	10	28	0.00	0.01	113	0.00	0.00
6	15	53	0.00	0.07	198	0.00	0.01
6	20	81	0.04	0.14	331	0.01	0.08
6	25	128	0.19	0.64	513	0.10	0.38
6	30	189	0.08	0.70	772	0.07	0.38
Average		90	0.05	0.22	361	0.02	0.11

Table 5: Results on Archetti et al. (2007) Instances with Low Inventory Holding Cost

		ALNS Fast Version			ALNS Slow Version		
$\|\mathcal{T}\|$	n	Runtime(s.)	Best Gap(%)	Avg Gap(%)	Runtime(s.)	Best Gap(%)	Avg Gap(%)
3	5	7	0.00	0.00	30	0.00	0.00
3	10	14	0.00	0.00	55	0.00	0.00
3	15	22	0.00	0.00	89	0.00	0.00
3	20	34	0.00	0.04	141	0.00	0.01
3	25	52	0.00	0.17	210	0.00	0.04
3	30	71	0.02	0.56	295	0.00	0.14
3	35	101	0.01	0.53	423	0.00	0.18
3	40	140	0.37	1.20	567	0.12	0.48
3	45	191	0.59	1.71	751	0.26	1.03
3	50	247	0.30	1.52	1009	0.25	1.00
6	5	13	0.00	0.00	54	0.00	0.00
6	10	28	0.00	0.02	109	0.00	0.01
6	15	49	0.00	0.03	188	0.00	0.00
6	20	77	0.08	0.26	315	0.05	0.15
6	25	121	0.25	1.29	493	0.24	0.65
6	30	182	0.67	1.90	726	0.07	1.06
Average		84	0.14	0.58	341	0.06	0.30

comparison, Archetti et al.'s (2012) algorithm obtains a gap of 0.06% and 0.10%, respectively. We are able to solve to optimality all instances with up to 35 customers for $|\mathcal{T}| = 3$, and with up to 15 customers for $|\mathcal{T}| = 6$. Similar quality results can also be found in Coelho et al. (2012a) and Coelho et al. (2012b), also when it comes to the higher gaps on the low inventory holding cost instances. A possible explanation could be that for low inventory holding costs the importance of the container selection decision in each period becomes relatively more pronounced. Our computation times are also very competitive compared to those in the literature, although a more rigorous scaling approach could be difficult due to the lack of precise processor architecture specifications in some of the works.

5.2.2 Results on VRP Benchmarks.

The SIRP that we study includes a rich routing problem. Since the routing component in the IRP benchmarks under consideration is very simple, we test our ALNS on two VRP benchmark instance sets, namely those of Crevier et al. (2007) and Taillard (1999).

Crevier et al. (2007) solve the Multi-depot VRP with Inter-depot routes (MDVRPI). Their instances consist of two sets of randomly generated instances with intermediate facilities, a homogeneous fixed fleet, and a maximum tour duration. Each vehicle's daily deployment cost φ_k and unit-time running cost θ_k are both equal to zero, and its unit-distance running cost $\beta_k = 1$. The set (a1–l1) includes 12 newly generated instances with two to five intermediate facilities and 48 to 216 customers. The set (a2–j2) includes 10 instances derived from those of Cordeau et al. (1997) by adding a central depot where the vehicles are stationed. It contains four to six intermediate facilities and 48 to 288 customers. The Best Known Solutions (BKS) to both sets are obtained by Hemmelmayr et al. (2013) who use a VNS with the dynamic programming procedure for the insertion of the intermediate facilities presented in Section 4.2. In Table 6, the instance name is followed by the computation time in seconds, the best and average cost obtained by Hemmelmayr et al. (2013) in 10 runs. The next three columns report the values produced by our ALNS. The last two columns represent the percent gap of our best and our average cost with respect to those of Hemmelmayr et al. (2013). Our best results are on average 0.49% from those of Hemmelmayr et al. (2013) and we are able to reach six of the BKS. The gap grows to about 1% for the average value over 10 runs.

Taillard (1999) formalizes the Heterogeneous Fixed Fleet VRP (HFFVRP). The version we solve, known as the HFFVRP with fixed and variable costs, considers a non-zero daily deployment cost φ_k and unit-distance running cost β_k, and a zero unit-time running cost θ_k. The instance set is derived from the eight largest Golden et al. (1984) instances by specifying φ_k and β_k for each vehicle k so that no single vehicle is better than any other in terms of its capacity to cost ratio. The instances include 50, 75, and 100 customers, three to six vehicle types and up to six vehicles per type. Taillard (1999) spurred a strong scientific interest in

Table 6: Results on Crevier et al. (2007) Instances

	Hemmelmayr et al. (2013)			ALNS				
Instance	Runtime(s.)	Best Cost	Avg Cost	Runtime(s.)	Best Cost	Avg Cost	Best Gap(%)	Avg Gap(%)
a2	73.80	997.94	997.94	108.49	997.94	998.17	0.00	0.02
b2	384.60	1291.19	1291.19	511.67	1291.19	1296.57	0.00	0.42
c2	900.60	1715.60	1715.84	1703.63	1718.34	1732.29	0.16	0.96
d2	1808.40	1856.84	1860.92	4833.00	1868.08	1887.63	0.61	1.44
e2	2958.60	1919.38	1922.81	10,288.66	1940.67	1954.59	1.11	1.65
f2	4274.40	2230.32	2233.43	20,968.55	2270.12	2296.05	1.78	2.80
g2	222.60	1152.92	1153.17	253.66	1152.92	1153.94	0.00	0.07
h2	939.60	1575.28	1575.28	1801.07	1582.64	1591.36	0.47	1.02
i2	2515.20	1919.74	1922.24	8062.43	1945.07	1972.93	1.32	2.64
j2	4402.80	2247.70	2250.21	19,591.55	2275.09	2295.72	1.22	2.02
a1	85.20	1179.79	1180.57	105.42	1179.79	1196.35	0.00	1.34
b1	383.40	1217.07	1217.07	575.90	1217.07	1222.49	0.00	0.44
c1	1224.00	1866.76	1867.96	3483.87	1876.57	1888.09	0.53	1.08
d1	94.20	1059.43	1059.43	97.97	1061.23	1062.27	0.17	0.27
e1	373.20	1309.12	1309.12	496.69	1309.12	1332.22	0.00	1.76
f1	1536.00	1570.41	1573.05	3998.06	1572.46	1589.58	0.13	1.05
g1	202.80	1181.13	1183.32	220.09	1185.35	1187.93	0.36	0.39
h1	876.60	1545.50	1548.61	1445.75	1555.70	1565.24	0.66	1.07
i1	2014.80	1922.18	1923.52	5504.89	1932.49	1946.75	0.54	1.21
j1	166.80	1115.78	1115.78	275.86	1118.84	1123.06	0.27	0.65
k1	873.60	1576.36	1577.96	1591.90	1582.32	1598.36	0.38	1.29
l1	2128.80	1863.28	1869.70	5992.99	1884.08	1909.46	1.12	2.13
Average	1292.73	1559.71	1561.32	4177.82	1568.96	1581.87	0.49	1.17

Table 7: Results on Taillard (1999) Instances

		ALNS				
Instance	BKS	Runtime(s.)	Best Cost	Avg Cost	Best Gap(%)	Avg Gap(%)
13	3185.09	359.52	3214.48	3236.09	0.92	1.60
14	10,107.53	378.82	10,140.11	10,148.63	0.32	0.41
15	3065.29	367.82	3069.76	3078.58	0.15	0.43
16	3265.41	365.38	3288.69	3306.93	0.71	1.27
17	2076.96	714.02	2109.72	2141.08	1.58	3.09
18	3743.58	780.14	3776.04	3835.14	0.87	2.45
19	10,420.34	1581.91	10,489.85	10,511.57	0.67	0.88
20	4761.26	1386.04	4896.63	4917.27	2.84	3.28
Average	5078.18	741.70	5123.16	5146.91	1.01	1.67

this problem resulting in at least a dozen algorithms in the literature. The proof of optimality of the solutions to the 50- and 75-customer instances of the problem is due to Baldacci and Mingozzi (2009). In Table 7, the instance name is followed by the BKS, which are due to multiple authors. Next are the computation time in seconds, the best and average cost obtained by our ALNS. The last two columns report the percent gap of our best and average cost with respect to the BKS. Our results are in the order of 1-2% from the BKS, most of which are proved to be optimal. Computation times are in the order of five to 25 minutes.

5.3 Case Study

In this section, we analyze the performance of our ALNS on sets of IRP instances derived from real data. Section 5.3.1 evaluates the effect of including the probability information in the objective function in terms of its impact on the expected cost and the frequency of occurrence of container overflows and route failures. Section 5.3.2 compares the probabilistic approach to alternative practical policies such as artificial buffer capacities at the containers and trucks. Section 5.3.3 employs a daily rolling horizon approach and derives empirical lower and upper bounds on the resulting cost over a given planning horizon.

5.3.1 Probabilistic policies.

The state-of-practice data used in Sections 5.3.1 and 5.3.2 includes 63 instances, each covering a week of white glass collections in the canton of Geneva, Switzerland in 2014, 2015, or 2016. Tours should respect a maximum duration of four hours each, and the time windows correspond to 8:00 a.m. to 12:00 p.m. The planning horizon is seven days long, starting on Monday and finishing on Sunday. As established by constraints (22) in the mathematical model, there should be no expected overflows on the first day after the planning horizon, in our case the following Monday. Each instance contains up to two heterogeneous vehicles of volume capacity in the order of 30,000 liters and weight capacity of 10,000 to 15,000 kg, which are not available on the weekend. On average, there are 41 containers per instance, and the maximum is 53, and their volumes range from 1000 to 3000 liters. There are two dumps located far apart in the periphery of the city of Geneva. For the trucks, we use a daily deployment cost of 100 CHF, a cost of 2.95 CHF per kilometer and a cost of 40 CHF per hour. The overflow cost, which is normally determined by the municipality, is set to 100 CHF. The demands for each instance are forecast by the model from Section 3.1 using the past 90 days of observations for each container. Two deposit sizes—two and ten liters—are used. For each instance, there is a distinct forecasting error υ estimated by formula (4).

We perform two types of experiments on these instances. The first one considers the complete objective function with all relevant costs, as defined by expression (9). We label the problem with this objective "Complete". The second one minimizes the routing cost defined by expression (11), ignoring all costs related to container overflows, emergency collections and route failures, and we label the problem with the latter objective "Routing-only". Tables 8, 9 and 10 summarize the numerical results. In these three tables, the first three columns identify the type of objective considered, the Emergency Collection Cost (ECC), and the Route Failure Cost Multiplier (RFCM). For each combination of emergency collection cost and RFCM, Table 8 reports the computation time and descriptive statistics about the average number of tours, container collections and dump visits, as well as the best and average cost for 10 runs and the percent gap between them. We observe that computation times are in the order of 10 to 15 minutes, which is acceptable for an operational problem that is solved on a daily basis. The results indicate clearly that the complete objective solution collects on average more than twice as many containers and, as a consequence, performs more tours and dump visits. In terms of expected cost, it is 50 to 60% more expensive. Since the optimal solution is not available, we can only judge the quality of the result based on the gap between the best and the average solution. It is in the order of 1% for the routing-only objective, and grows to roughly 2% for the complete objective, reflecting the more challenging search space produced

Table 8: Probabilistic Policies: Basic Results for Cost Analysis on Real Data Instances

Objective	ECC	RFCM	Runtime(s.)	Avg Num Tours	Avg Num Containers	Avg Num Dump Visits	Best Cost (CHF)	Avg Cost (CHF)	Gap Avg-Best(%)
Complete	100.00	1.00	781.71	1.96	43.44	2.31	664.76	679.54	2.22
Complete	100.00	0.50	862.13	1.96	43.43	2.30	664.82	678.84	2.11
Complete	100.00	0.25	806.52	1.95	43.52	2.28	664.34	677.81	2.03
Complete	100.00	0.00	715.82	1.95	43.80	2.28	664.00	677.11	1.97
Complete	50.00	1.00	915.61	1.92	41.08	2.20	650.86	662.18	1.74
Complete	50.00	0.50	812.67	1.91	41.22	2.21	650.55	662.28	1.80
Complete	50.00	0.25	809.76	1.91	41.19	2.19	650.72	661.88	1.71
Complete	50.00	0.00	790.21	1.91	41.07	2.19	651.09	661.93	1.66
Complete	25.00	1.00	814.44	1.90	39.56	2.13	641.43	651.24	1.53
Complete	25.00	0.50	789.00	1.90	39.56	2.14	641.79	652.04	1.60
Complete	25.00	0.25	789.40	1.90	39.57	2.15	641.42	651.85	1.63
Complete	25.00	0.00	783.33	1.89	39.59	2.13	642.71	651.71	1.40
Routing-only	0.00	0.00	725.46	1.83	16.77	1.87	422.64	425.08	0.58

by its non-linear objective function. It appears that the solution cost is strongly influenced by the emergency collection cost, but almost unaffected by the RFCM.

Table 9 is a more detailed breakdown of the cost and efficiency structure of the various policies. The fourth, fifth and sixth column decompose the average solution cost from Table 8 into routing, overflow and route failure cost. The last three columns report the total collected volume in liters, and the volume per unit of total cost and routing cost, which can be regarded as performance indicators. The results reveal that the routing cost of the complete objective solution is on average only 30 to 35% higher than that of the routing-only objective solution. The rest of the difference in the total solution cost is explained by the contribution of the expected overflow cost. The routing cost is lower for a lower emergency collection cost, while the expected overflow cost is higher. A higher emergency collection cost necessitates more frequent visits as an attempt to further limit overflows. On the contrary, paying a higher emergency collection cost for a lower number of resulting overflows in this case does slightly reduce the expected overflow cost captured in the objective function. The route failure cost in both solutions is practically null. Not surprisingly, the solutions with the complete objective collect more volume as well. However, a better indication of their efficiency is provided by the collected volume per unit cost, which is 15% higher with respect to the total cost, and 35% higher with respect to the routing cost.

The relevance of the probability information captured by the objective function can be evaluated through the analysis of the occurrence of extreme events. After solving each instance, we perform 10,000 simulations. The forecasting error is sampled independently for each container and each day using the estimate υ. We then evaluate the effect on the occurrence of container overflows and route failures in the solution provided by the ALNS algorithm. An overflow is counted on each day, i.e. if a container is overflowing on two consecutive days because it is not collected, we count two overflow events. Table 10 summarizes the average number of overflows and route failures at the 75th, 90th, 95th and 99th percentiles of the 10,000 simulation runs on each of the 63 instances. We observe a strong negative correlation of the average number of overflows with the emergency collection cost and of the average number of route failures with

Table 9: Prob. Policies: Key Performance Indicators for Cost Analysis on Real Data Instances

Objective	ECC	RFCM	Avg Routing Cost (CHF)	Avg Overflow Cost (CHF)	Avg Rte Failure Cost (CHF)	Avg Collected Volume (L)	Liters Per Unit Cost	Liters Per Unit Routing Cost
Complete	100.00	1.00	579.78	99.73	0.03	47,234.59	69.51	81.47
Complete	100.00	0.50	579.46	99.33	0.05	47,225.62	69.57	81.50
Complete	100.00	0.25	577.84	99.93	0.04	47,455.19	70.01	82.13
Complete	100.00	0.00	578.83	98.28	0.00	47,662.90	70.39	82.34
Complete	50.00	1.00	559.44	102.72	0.02	45,646.48	68.93	81.59
Complete	50.00	0.50	558.37	103.82	0.09	45,852.89	69.24	82.12
Complete	50.00	0.25	558.47	103.35	0.07	45,949.94	69.42	82.28
Complete	50.00	0.00	557.16	104.77	0.00	45,788.15	69.17	82.18
Complete	25.00	1.00	547.74	103.46	0.04	44,682.00	68.61	81.57
Complete	25.00	0.50	548.10	103.83	0.11	44,653.66	68.48	81.47
Complete	25.00	0.25	547.75	104.05	0.06	44,678.38	68.54	81.57
Complete	25.00	0.00	546.34	105.37	0.00	44,773.34	68.70	81.95
Routing-only	0.00	0.00	425.08	0.00	0.00	25,286.94	59.49	59.49

Table 10: Prob. Policies: Container Overflows and Route Failures for Real Data Instances

			Avg Num Overflows				Avg Num Route Failures			
Objective	ECC	RFCM	75th Perc.	90th Perc.	95th Perc.	99th Perc.	75th Perc.	90th Perc.	95th Perc.	99th Perc.
Complete	100.00	1.00	0.98	1.78	2.40	3.58	0.03	0.03	0.04	0.05
Complete	100.00	0.50	0.99	1.78	2.39	3.55	0.04	0.05	0.05	0.07
Complete	100.00	0.25	0.97	1.80	2.38	3.56	0.04	0.05	0.06	0.10
Complete	100.00	0.00	0.94	1.77	2.33	3.54	0.08	0.10	0.12	0.16
Complete	50.00	1.00	1.26	2.19	2.82	4.14	0.05	0.05	0.05	0.05
Complete	50.00	0.50	1.28	2.19	2.84	4.16	0.06	0.07	0.08	0.09
Complete	50.00	0.25	1.28	2.18	2.83	4.15	0.04	0.06	0.07	0.10
Complete	50.00	0.00	1.31	2.23	2.85	4.18	0.07	0.09	0.10	0.12
Complete	25.00	1.00	1.48	2.46	3.14	4.58	0.05	0.05	0.05	0.07
Complete	25.00	0.50	1.48	2.46	3.14	4.58	0.05	0.07	0.07	0.10
Complete	25.00	0.25	1.51	2.50	3.18	4.61	0.04	0.07	0.07	0.09
Complete	25.00	0.00	1.54	2.51	3.19	4.64	0.08	0.10	0.10	0.12
Routing-only	0.00	0.00	16.97	20.45	22.56	26.70	0.01	0.03	0.04	0.05

the RFCM. What is more striking, however, is the difference between the series of complete solutions on the one hand and the routing-only solution on the other. While the complete solutions are able to limit the number of overflows to no more than five, even at the extreme of the simulated distribution, the average number of overflows in the routing-only solution is higher by a degree of magnitude.

Figure 4 is a visual representation of the average cost of overflows that the collector would pay at the 75th, 90th, 95th and 99th percentile of the simulated demand distribution over all 63 instances, for the routing only solution and for the complete solution with an ECC of 100 CHF and an RFCM equal to one. The differences are consequential. The cost due to the routing-only objective is from 17 times higher at the 75th percentile to 7 times higher at the 99th percentile, which is a clear indication of the underestimation of risk in the face of stochastic demand. Even at the 99th percentile, the complete objective would result in a total cost of less than 1000 CHF, compared to more than 3000 CHF for the routing-only objective.

Figures 5 depicts the average number of container overflows at the 75th, 90th, 95th and 99th percentile for each instance over 10 runs. For the complete objective with an ECC of 100 CHF

Figure 4: Average Cost of Overflows at Different Percentiles of the Simulated Distribution

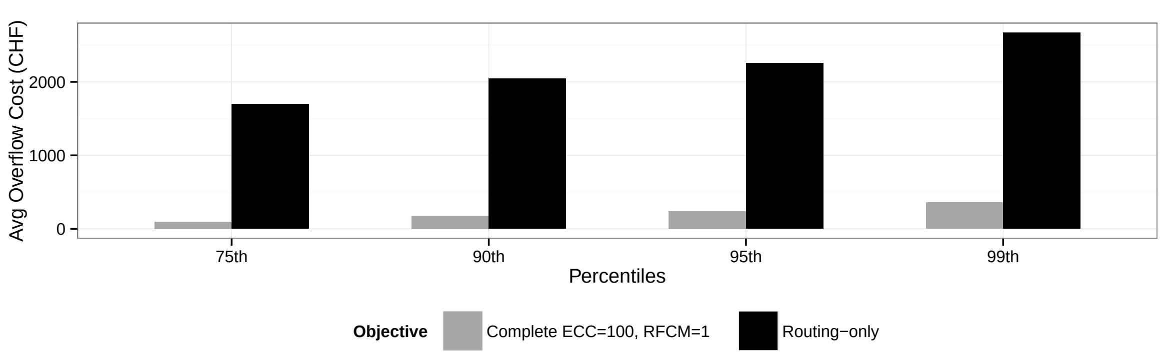
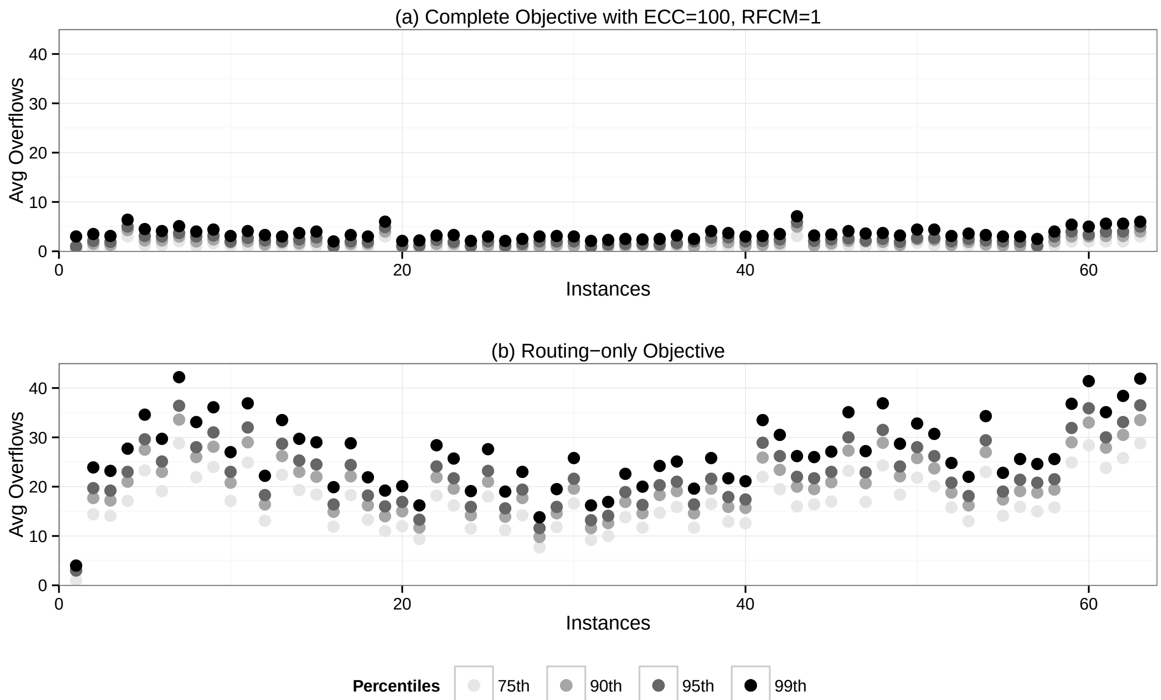


Figure 5: Overflows for All Instances at Different Percentiles of the Simulated Distribution



and an RFCM equal to one, the ratio of container overflows to the number of containers in the instance goes from 2% at the 75th to 10% at the 99th percentile. For the routing-only objective, these values are 41% and 66%, respectively.

To study the main drivers of the number of container overflows, we perform a series of linear regressions. Table 11 consists of two parts. In part (a) the explanatory factor is the forecasting error υ, while in part (b) it is the number of containers in the instance. In both parts, the first column identifies the type of objective considered, and the rest of the columns correspond to the dependent variable, i.e. the average number of container overflows at the 75th, 90th, 95th and 99th percentile. For each of them, we report the coefficient of the explanatory factor followed by a significance code, and the coefficient of determination R^2. We observe that all coefficients are positive as expected and most of them are significant at the 99% confidence level. The regressions on the forecasting error suggest that it explains approximately half of the variability in the container overflows for the routing-only objective and about a third in the case of the complete objective with an ECC of 100 CHF and an RFCM equal to one. This result is intuitive as higher forecasting errors lead to larger demand perturbations in the simulation experiments and, as a consequence, to a higher rate of overflows. The results of the regressions on the number of containers in the instance exhibit a more pronounced difference. While it can explain approximately 25% of the variability in the container overflows for the routing-only objective, the number of containers in the instance seems not to have an effect on the overflows in the case of the complete objective with an ECC of 100 CHF and an RFCM equal to one.

Table 11: Driving Factors for the Occurrence of Container Overflows

(a) Regressions on Forecasting Error υ

	75th Percentile		90th Percentile		95th Percentile		99th Percentile	
Objective	Coefficient	R^2	Coefficient	R^2	Coefficient	R^2	Coefficient	R^2
Complete with ECC=100, RFCM=1	0.02***	0.36	0.02***	0.31	0.03***	0.37	0.03***	0.39
Routing-only	0.16***	0.52	0.18***	0.52	0.19***	0.51	0.21***	0.51

(b) Regressions on Number of Containers in the Instance

	75th Percentile		90th Percentile		95th Percentile		99th Percentile	
Objective	Coefficient	R^2	Coefficient	R^2	Coefficient	R^2	Coefficient	R^2
Complete with ECC=100, RFCM=1	0.03**	0.08	0.03**	0.07	0.03**	0.07	0.04**	0.07
Routing-only	0.34***	0.25	0.37***	0.24	0.41***	0.26	0.47***	0.27

Note: Significance codes *** 99%, ** 95%

We observe both very low values of the R^2 statistic and less significant coefficients. This is a desirable result as it would suggest that the number of overflows does not scale up with the instance size. It also has a managerial implication, giving a reliable estimate of extreme events over a wide range of situations.

5.3.2 Alternative Policies.

To further study the theoretical justification and practical relevance of the probabilistic approach, we compare it to an intuitive routing-only approach, in which during the solution of the problem we use artificially low capacities for the containers and the trucks. This policy is an attempt to control the number of container overflows and route failures and it also leads, undoubtedly, to higher routing costs due to the necessity of more frequent visits. After each instance is solved, we perform the same simulation-based validation of the solution as in Section 5.3.1. However, during the simulation we count the number of container overflows and route failures with respect to the original container and truck capacities. Thus, we have a fair comparison between the probabilistic approach and the alternative policies of artificially low capacities.

Tables 12, 13 and 14 are structured in the same way as Tables 8, 9 and 10 in Section 5.3.1. Here, the objective is always routing-only and what varies are the Container Effective Capacity (CEC) and the Truck Effective Capacity (TEC) as fractions of their original capacities. In Table 12, we note the strong negative correlation between the container effective capacity and the average number of tours, collected containers and dump visits in the solutions. We also notice that the relative increase in the number of collected containers is much higher than the reduction of the container effective capacity. This is an artifact of the finite planning horizon as many containers may be collected two or three times rather than once or twice due to their smaller effective capacities. This effect will most likely diminish over the long run. The solution cost appears to be influenced by both the container and the truck effective capacity.

Interestingly, the solution gap grows substantially when the container effective capacity is reduced to 75%. One explanation could be that since demands on certain days are quite high, this configuration leads to a tighter and at the same time more fragmented search space of feasible solutions for the ALNS.

Table 13 shows the gradual growth of the routing cost as we reduce the effective capacities. Since the objective is always routing-only, the overflow and route failure components do not apply. The last three columns show that, in addition to collecting more containers and as a result more volume, the solutions with lower container effective capacities also collect more volume per unit cost and unit routing cost. This is easily explained by the fact that collecting more containers in the same geographic area reduces the average distance among them while at the same time increasing the total collected volume. Table 14 describes the average results of the 10,000 simulation runs that were performed on each instance with the original container and truck effective capacities. It is immediately clear that considering artificially low capacities during the solution has a marked effect in reducing overflows and route failures. To be precise, the average number of overflows drops by roughly a third when the container effective capacity is reduced to 90% and by roughly two thirds when it is reduced to 75%. On the other hand, reducing the truck effective capacity to 90% can effectively eliminate the occurrence of route failures.

Figures 6 and 7 present a side-by-side comparison of the probabilistic and the alternative poli-

Table 12: Alternative Policies: Basic Results for Cost Analysis on Real Data Instances

Objective	CEC	TEC	Runtime(s.)	Avg Num Tours	Avg Num Containers	Avg Num Dump Visits	Best Cost (CHF)	Avg Cost (CHF)	Gap Avg-Best(%)
Routing-only	1.00	1.00	812.43	1.83	16.77	1.87	422.72	425.48	0.65
Routing-only	1.00	0.90	845.99	1.84	16.72	1.88	422.73	426.94	0.99
Routing-only	1.00	0.75	865.26	1.83	16.81	1.93	424.29	428.02	0.88
Routing-only	0.90	1.00	882.96	2.00	22.69	2.04	486.88	488.76	0.39
Routing-only	0.90	0.90	853.53	2.00	22.69	2.06	487.38	489.20	0.37
Routing-only	0.90	0.75	860.20	2.00	22.71	2.17	489.55	491.91	0.48
Routing-only	0.75	1.00	1003.83	2.10	33.80	2.57	547.48	564.83	3.17
Routing-only	0.75	0.90	1010.03	2.11	33.87	2.73	553.27	570.32	3.08
Routing-only	0.75	0.75	1010.74	2.11	33.89	2.97	558.16	575.98	3.19

Table 13: Alt. Policies: Key Performance Indicators for Cost Analysis on Real Data Instances

Objective	CEC	TEC	Avg Routing Cost (CHF)	Avg Overflow Cost (CHF)	Avg Rte Failure Cost (CHF)	Avg Collected Volume (L)	Liters Per Unit Cost	Liters Per Unit Routing Cost
Routing-only	1.00	1.00	425.48	0.00	0.00	25,311.81	59.49	59.49
Routing-only	1.00	0.90	426.94	0.00	0.00	25,233.43	59.10	59.10
Routing-only	1.00	0.75	428.02	0.00	0.00	25,371.43	59.28	59.28
Routing-only	0.90	1.00	488.76	0.00	0.00	31,532.12	64.51	64.51
Routing-only	0.90	0.90	489.20	0.00	0.00	31,611.40	64.62	64.62
Routing-only	0.90	0.75	491.91	0.00	0.00	31,732.72	64.51	64.51
Routing-only	0.75	1.00	564.83	0.00	0.00	44,134.12	78.14	78.14
Routing-only	0.75	0.90	570.32	0.00	0.00	44,084.86	77.30	77.30
Routing-only	0.75	0.75	575.98	0.00	0.00	44,079.24	76.53	76.53

Table 14: Alt. Policies: Container Overflows and Route Failures for Real Data Instances

			Avg Num Overflows				Avg Num Route Failures			
Objective	CEC	TEC	75th Perc.	90th Perc.	95th Perc.	99th Perc.	75th Perc.	90th Perc.	95th Perc.	99th Perc.
Routing-only	1.00	1.00	16.97	20.45	22.58	26.72	0.01	0.03	0.03	0.04
Routing-only	1.00	0.90	17.02	20.51	22.65	26.80	0.00	0.00	0.00	0.00
Routing-only	1.00	0.75	16.91	20.40	22.54	26.65	0.00	0.00	0.00	0.00
Routing-only	0.90	1.00	10.32	13.14	14.85	18.29	0.02	0.02	0.02	0.02
Routing-only	0.90	0.90	10.30	13.09	14.81	18.24	0.00	0.00	0.00	0.00
Routing-only	0.90	0.75	10.32	13.09	14.85	18.28	0.00	0.00	0.00	0.00
Routing-only	0.75	1.00	4.24	6.08	7.27	9.68	0.03	0.03	0.03	0.03
Routing-only	0.75	0.90	4.24	6.06	7.26	9.68	0.00	0.00	0.00	0.00
Routing-only	0.75	0.75	4.22	6.04	7.26	9.67	0.00	0.00	0.00	0.00

cies of using artificially low container and truck capacities. In both figures, the first 12 bars represent the probabilistic model with complete objective function for various Emergency Collection Costs (ECC) and Route Failure Cost Multipliers (RFCM). The last nine bars represent the alternative policies of using artificially low capacity for various Container Effective Capacities (CEC) and Truck Effective Capacities (TEC). We should point out that the baseline routing-only policy with container and truck effective capacity of 100% has the lowest routing cost. Figure 6 reveals that the routing cost of the probabilistic policies considered ranges from approximately 550 to 580 CHF depending mostly on the value of the emergency collection cost. This latter range is relatively limited compared to the range of routing costs for the alternative policies, which goes from 425 to 575 CHF, with pronounced jumps linked to the variation of the container effective capacity. Thus the most expensive probabilistic and the most expensive alternative policies that we consider have basically the same routing cost.

We contrast the above observation with the average number of overflows and route failures after the simulation-based validation of both types of policies. These are presented in Figure 7, in parts (a) and (b), respectively. What part (a) of the figure reveals is that all considered probabilistic policies are able to contain the number of overflows to very low values. There is still a slight increase in the number of overflows (with an associated slight decrease in the routing cost) when the emergency collection cost is reduced from 100 to 50 and then to 25 CHF.

Figure 6: Comparison of Routing Cost for Probabilistic and Alternative Policies

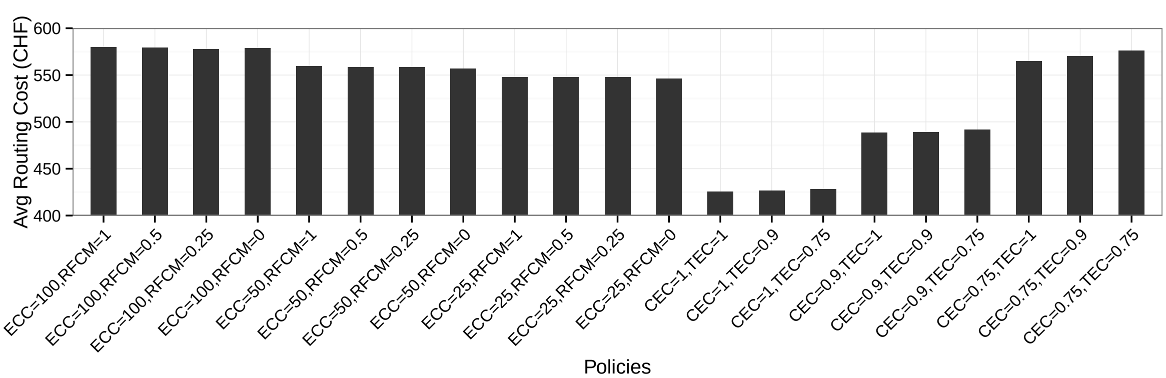
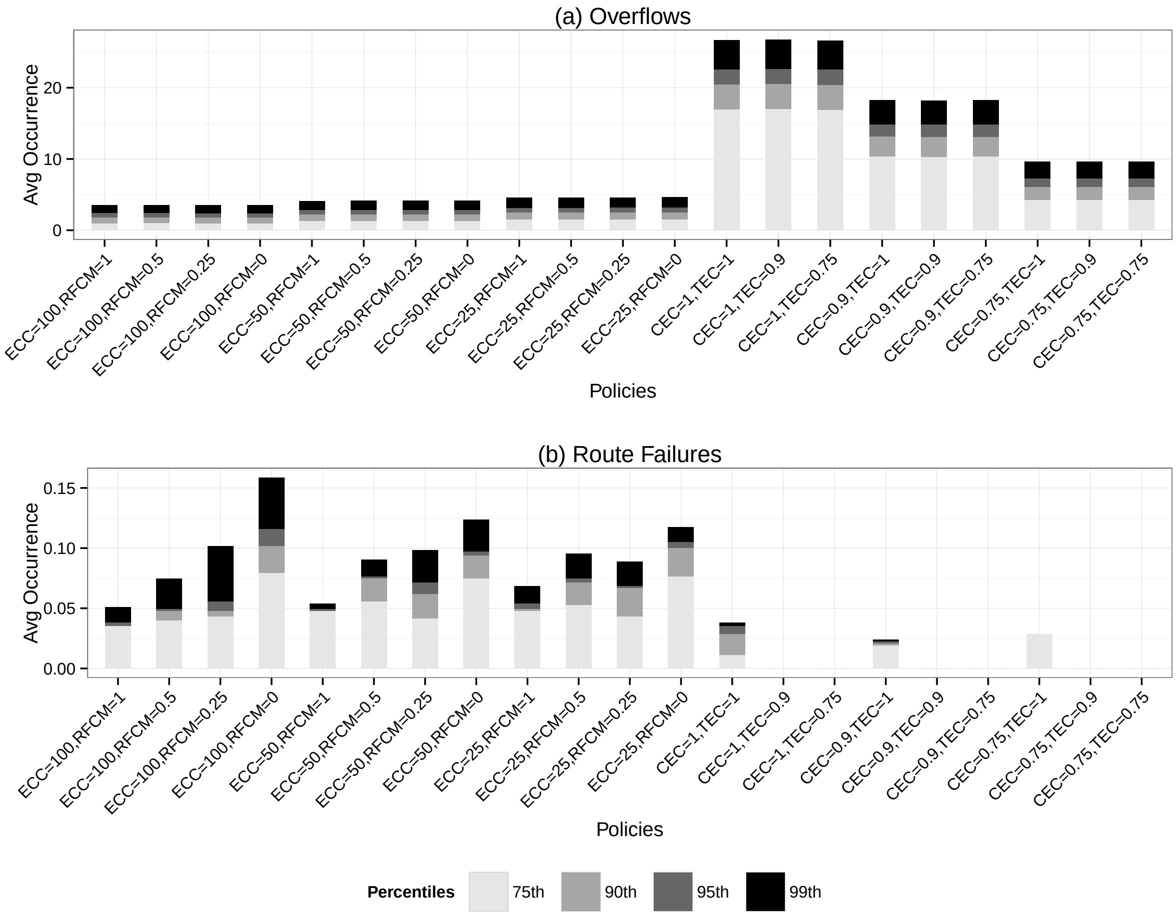


Figure 7: Comparison of Container Overflows and Route Failures for Prob. and Alt. Policies



Nevertheless, the average number of overflows across all instances is less than five even at the 99th percentile. In comparison, the average number of overflows for the alternative policies is markedly higher. While reducing the container effective capacity leads to a considerable drop in the number of overflows, the best values are still more than twice as high as those for the probabilistic policies with similar or lower routing cost. We must stress here that since we compare the performance of two policy types in terms of number of overflows at different percentiles, we must isolate that component from the solution cost of the probabilistic model. Since the route failure cost is a negligible component, the comparison of the routing costs provides an excellent benchmark. The above findings clearly indicate the superior performance of the probabilistic policies in the face of stochastic demand. Whereas the alternative policies can only control overflows in the expected sense, the probabilistic model attributes a cost to this uncertainty over the whole planning horizon. Thus it uses foresight in a much more intelligent way. Not surprisingly, it collects more containers over the planning horizon for an almost identical routing cost.

Lastly, part (b) of Figure 7 shows how both types of policies perform in terms of the average number of route failures over all instances. Here, the alternative policies appear to be more successful. As already noted before, reducing the truck effective capacity to 90% is sufficient to eliminate the occurrence of route failures. As far as the probabilistic policies are concerned, we identify an interesting pattern. The number of route failures is positively correlated with the emergency collection cost and negatively correlated with the route failure cost multiplier. The latter is an intuitive result. The former relationship, however, is slightly more intricate. What is at play here is a trade-off between container overflows and route failures. A higher emergency collection cost incentivizes more frequent visits. Trucks thus collect more containers in each tour and, by consequence, in each depot-to-dump or dump-to-dump visit. Since trucks are fuller on average, the solution is subject to a higher risk of route failures. The probabilistic policies collect on average more containers than the alternative policies and this could be a valid explanation of the latter's better performance when it comes to limiting the number of route failures. However, as reported in Table 9, the contribution of the expected route failure cost to the total solution cost is immaterial.

5.3.3 Rolling Horizon Approach.

In practice, the SIRP that we consider will be solved on a daily rolling horizon basis using the latest available container level information. In this approach, the problem is solved for a planning horizon $\mathcal{T}$, the tours that are scheduled on day $t = 0$ are executed, the horizon is rolled over by a day, the problem is re-solved for the new initial container levels and updated forecasts, and so on. Thus, true demands are gradually revealed each day, but the demands over the planning horizon are still stochastic. This type of problem is known as the Dynamic and Stochastic Inventory Routing Problem (DSIRP). The solution of the DSIRP requires the solution of an SIRP at each rollover. The cost of the DSIRP is composed of the total routing and overflow cost on day $t = 0$ resulting from the solution of the SIRP at each rollover. We note that the route failure cost does not apply on day $t = 0$. We also note that overflows on day $t = 0$ are deterministic, since the container levels are fully known, and thus for each overflow on day $t = 0$ the full overflow cost χ is paid.

In the solution of the DSIRP, true demands are gradually revealed in the solution process, which reduces uncertainty. Thus, we hypothesize that its solution cost should be bounded above by the solution cost of a static SIRP for the same planning horizon. Assume that we solve the SIRP for a planning horizon $\mathcal{T} = \{0, \dots, u\}$. In order to compare its cost to that of the DSIRP, we should roll over for a number of times equal to the length of the planning horizon $\mathcal{T}$, i.e. the last rollover should be on day u. Moreover, for rollover t the initial container levels are updated by true demands and also dependent on the solution of rollover $t - 1$. Updated forecasts should ideally be used if available. We also hypothesize that the solution of the DSIRP should be bounded below by the solution of a static IRP using true demands for the same planning horizon $\mathcal{T}$. Using true demands rids the problem of any uncertainty. The solution of the IRP results in an intelligent assignment of tours to days. Thus, the number of executed tours over the planning horizon will be minimized and tours may not be executed on each day. This is not necessarily the case for the solution of the DSIRP, which has no memory of the past rollovers and may assign tours on day $t = 0$ for each rollover.

To test our hypotheses, we generate a second set of real data instances. It comprises 41 instances, each covering two weeks of white glass collections in the canton of Geneva, Switzerland in 2014, 2015, or 2016. The instances fit the same description as the previous set of 63 instances. On average, there are 69 containers per instance, and the maximum is 86. We solve the static IRP with true demands and static SIRP with forecast demands for the first week, and the DSIRP with a one week planning horizon and rollovers for the first week. Table 15 presents the results we obtain. Since we are interested in verifying the empirical existence of the hypothesized bounds, we report the best cost out of 10 runs. We observe that the hypothesized bounds are obtained in all but four cases, which are shown in bold. The relative differences are also very interesting to look at. The solutions of the DSIRP are on average 60% more expensive than those of the static IRP with true demands. This result is inevitably related to the level of uncertainty as represented by the forecasting error υ. In other words, if more accurate forecasting methodologies are available, this gap may be brought down. On the other hand, the static SIRP approach is on average 15% more expensive than the rollover approach for the DSIRP. This clearly shows the benefit of applying the latter in practical situations.

6 Conclusion

We motivate and formulate a rich stochastic IRP inspired from practice. Our objective function captures the routing cost and the relevant uncertain components with the goal of minimizing

Table 15: Analysis of Rolling Horizon DSIRP Bounds

Instance	Static IRP with True Demand	Rolling DSIRP with Forecast Demand	Static SIRP with Forecast Demand	Instance	Static IRP with True Demand	Rolling DSIRP with Forecast Demand	Static SIRP with Forecast Demand
Inst_1	276.44	582.89	665.19	Inst_22	429.20	531.04	607.63
Inst_2	448.67	784.55	854.49	Inst_23	241.44	551.58	690.62
Inst_3	307.95	653.60	819.79	**Inst_24**	**547.92**	**758.84**	**748.71**
Inst_4	266.15	574.23	700.36	Inst_25	446.31	618.80	696.75
Inst_5	454.61	682.24	824.57	Inst_26	442.38	589.53	695.11
Inst_6	485.30	677.92	764.86	Inst_27	441.36	589.07	707.30
Inst_7	268.65	569.11	649.57	Inst_28	468.46	616.53	738.58
Inst_8	429.56	585.42	681.23	Inst_29	436.25	575.25	701.73
Inst_9	442.34	599.24	659.30	Inst_30	414.41	677.65	690.37
Inst_10	448.70	564.04	650.88	Inst_31	442.87	544.75	668.51
Inst_11	467.88	549.61	670.36	Inst_32	255.32	612.44	694.35
Inst_12	**449.20**	**674.53**	**626.18**	Inst_33	460.04	677.54	808.74
Inst_13	254.66	556.94	629.93	Inst_34	505.55	682.90	711.62
Inst_14	276.60	585.77	683.65	**Inst_35**	**490.37**	**989.21**	**785.51**
Inst_15	431.08	548.56	790.39	Inst_36	454.60	646.95	805.95
Inst_16	529.60	635.37	701.64	Inst_37	465.31	607.52	746.64
Inst_17	423.07	578.84	662.76	Inst_38	520.38	721.23	815.21
Inst_18	458.18	595.36	680.75	Inst_39	243.94	613.96	705.10
Inst_19	448.66	524.63	611.56	Inst_40	450.94	624.76	759.97
Inst_20	418.12	520.30	653.18	Inst_41	403.01	575.80	688.24
Inst_21	**276.32**	**791.63**	**626.29**				

Note: The four instances for which the hypothesized bounds do not hold are shown in bold.

the expected cost subject to a range of practical and policy-related constraints. To solve the problem, we develop an ALNS algorithm which produces excellent results on IRP benchmarks sets from the literature, as well as very good results on rich vehicle routing instances. The application of the methodology to instances derived from real data demonstrates the relevance of the probability information captured in the objective function. The computational experiments demonstrate that including probabilistic information in the objective function leads to only a moderate increase in the routing cost, while avoiding major expenditures even at relatively lower percentiles of the demand distribution. Based on our policy, we can control the rate of occurrence of undesirable events, like overflows and route failures, by scaling the probability-related costs considered in the objective function. The probabilistic approach significantly outperforms alternative policies of using artificially low capacities for the containers and the trucks in its ability to control the occurrence of container overflows for the same routing cost. We also analyze the solution properties of a rolling horizon approach and propose empirical lower and upper bounds. Given that our problem is interesting from both a theoretical and a practical point of view, it lends itself to a rich variety of potential future work directions. Decomposition methods such as column generation may be investigated for calculating optimal solutions on small and medium-size instances or lower bounds on larger instances with the purpose of providing more meaningful benchmark results for the ALNS. Scenario-generation and the integration of chance constraints present another research avenue, oriented more towards the modeling approach rather than the solution methodology. More practically relevant ideas include the integration of a location aspect regarding the dumps, the possibility of open tours, online re-optimization and the solution of a multi-flow problem.

Acknowledgment

The work of Iliya Markov and Sacha Varone is funded by Switzerland's Commission for Technology and Innovation (CTI) under grant number CTI 15781.1 PFES-ES. This support is gratefully acknowledged. The authors thank EcoWaste SA, industrial partners under this grant, for their collaboration, expert advice and discussion on industry and problem specific issues, and the data they provided for the experiments in this study. The authors would also like to thank Raphaël Lüthi, a Master's thesis student at the Transport and Mobility Laboratory of École Polytechnique Fédérale de Lausanne, for his dedicated work on this problem, and Matthieu de Lapparent, a scientific collaborator at the same laboratory, for his advice on probability and statistics related questions.

References

Abdelmaguid, T. F. (2004). *Heuristic approaches for the integrated inventory distribution problem.* PhD thesis, University of Southern California, Los Angeles, CA, USA.

Abdollahi, M., Arvan, M., Omidvar, A., and Ameri, F. (2014). A simulation optimization approach to apply value at risk analysis on the inventory routing problem with backlogged demand. *International Journal of Industrial Engineering Computations*, 5(4):603–620.

Abramowitz, M. and Stegun, I. A., editors (1972). *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. New York: Dover Publications.

Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2015). Benders decomposition for production routing under demand uncertainty. *Operations Research*, 63(4):851–867.

Andersson, H., Hoff, A., Christiansen, M., Hasle, G., and Løkketangen, A. (2010). Industrial aspects and literature survey: Combined inventory management and routing. *Computers & Operations Research*, 37(9):1515–1536.

Angelelli, E. and Speranza, M. G. (2002a). The application of a vehicle routing model to a waste-collection problem: Two case studies. *The Journal of the Operational Research Society*, 53(9):944–952.

Angelelli, E. and Speranza, M. G. (2002b). The periodic vehicle routing problem with intermediate facilities. *European Journal of Operational Research*, 137(2):233–247.

Archetti, C., Bertazzi, L., Hertz, A., and Speranza, M. G. (2012). A hybrid heuristic for an inventory routing problem. *INFORMS Journal on Computing*, 24(1):101–116.

Archetti, C., Bertazzi, L., Laporte, G., and Speranza, M. G. (2007). A branch-and-cut algorithm for a vendor-managed inventory-routing problem. *Transportation Science*, 41(3):382–391.

Baldacci, R. and Mingozzi, A. (2009). A unified exact method for solving different classes of vehicle routing problems. *Mathematical Programming*, 120(2):347–380.

Bard, J. F., Huang, L., Dror, M., and Jaillet, P. (1998a). A branch and cut algorithm for the VRP with satellite facilities. *IIE Transactions*, 30(9):821–834.

Bard, J. F., Huang, L., Jaillet, P., and Dror, M. (1998b). A decomposition approach to the inventory routing problem with satellite facilities. *Transportation Science*, 32(2):189–203.

Beltrami, E. J. and Bodin, L. D. (1974). Networks and vehicle routing for municipal waste collection. *Networks*, 4(1):65–94.

Bertazzi, L., Bosco, A., Guerriero, F., and Laganà, D. (2013). A stochastic inventory routing problem with stock-out. *Transportation Research Part C: Emerging Technologies*, 27:89–107.

Bertazzi, L., Bosco, A., and Laganà, D. (2015). Managing stochastic demand in an inventory routing problem with transportation procurement. *Omega*, 56:112–121.

Bertsimas, D. and Sim, M. (2003). Robust discrete optimization and network flows. *Mathematical Programming*, 98(1-3):49–71.

Bertsimas, D. and Sim, M. (2004). The price of robustness. *Operations Research*, 52(1):35–53.

Bitsch, B. (2012). Inventory routing with stochastic demand. Master's thesis, Aarhus School of Business and Social Sciences, Aarhus University, Aarhus, Denmark.

Buhrkal, K., Larsen, A., and Ropke, S. (2012). The waste collection vehicle routing problem with time windows in a city logistics context. *Procedia-Social and Behavioral Sciences*, 39:241–254.

Campbell, A. M. and Savelsbergh, M. W. P. (2004). A decomposition approach for the inventory-routing problem. *Transportation Science*, 38(4):488–502.

Coelho, L. C., Cordeau, J.-F., and Laporte, G. (2012a). Consistency in multi-vehicle inventory-routing. *Transportation Research Part C: Emerging Technologies*, 24:270–287.

Coelho, L. C., Cordeau, J.-F., and Laporte, G. (2012b). The inventory-routing problem with transshipment. *Computers & Operations Research*, 39(11):2537–2548.

Coelho, L. C., Cordeau, J.-F., and Laporte, G. (2014a). Heuristics for dynamic and stochastic inventory-routing. *Computers & Operations Research*, 52(A):55–67.

Coelho, L. C., Cordeau, J.-F., and Laporte, G. (2014b). Thirty years of inventory routing. *Transportation Science*, 48(1):1–19.

Conrad, R. G. and Figliozzi, M. A. (2011). The recharging vehicle routing problem. In Doolen, T. and Aken, E. V., editors, *Proceedings of the 2011 Industrial Engineering Research Conference*, Reno, NV, USA.

Cordeau, J.-F., Gendreau, M., and Laporte, G. (1997). A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, 30(2):105–119.

Crevier, B., Cordeau, J.-F., and Laporte, G. (2007). The multi-depot vehicle routing problem with inter-depot routes. *European Journal of Operational Research*, 176(2):756–773.

Dror, M. and Ball, M. (1987). Inventory/routing: Reduction from an annual to a short-period problem. *Naval Research Logistics*, 34(6):891–905.

Dror, M., Ball, M., and Golden, B. (1985). A computational comparison of algorithms for the inventory routing problem. *Annals of Operations Research*, 4(1):1–23.

Dror, M. and Levy, L. (1986). A vehicle routing improvement algorithm comparison of a "greedy" and a matching implementation for inventory routing. *Computers & Operations Research*, 13(1):33–45.

Dror, M. and Trudeau, P. (1996). Cash flow optimization in delivery scheduling. *European Journal of Operational Research*, 88(3):504–515.

Erdoğan, S. and Miller-Hooks, E. (2012). A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 48(1):100–114.

European Commission (2016). *Circular Economy Strategy.* http://ec.europa.eu/environment/circular-economy.

Goeke, D. and Schneider, M. (2015). Routing a mixed fleet of electric and conventional vehicles. *European Journal of Operational Research*, 245(1):81–99.

Golden, B., Assad, A., Levy, L., and Gheysens, F. (1984). The fleet size and mix vehicle routing problem. *Computers & Operations Research*, 11(1):49–66.

Greco, G., Allegrini, M., Lungo, C. D., Savellini, P. G., and Gabellini, L. (2015). Drivers of solid waste collection costs. Empirical evidence from Italy. *Journal of Cleaner Production*, 106:364–371.

Hemmelmayr, V., Doerner, K. F., Hartl, R. F., and Rath, S. (2013). A heuristic solution method for node routing based solid waste collection problems. *Journal of Heuristics*, 19(2):129–156.

Hemmelmayr, V., Doerner, K. F., Hartl, R. F., and Savelsbergh, M. W. (2010). Vendor managed inventory for environments with stochastic product usage. *European Journal of Operational Research*, 202(3):686–695.

Hiermann, G., Puchinger, J., and Hartl, R. F. (2014). The electric fleet size and mix vehicle routing problem with time windows and recharging stations. Working paper, Austrian Institute of Technology and University of Vienna, Austria.

Ivarsøy, K. S. and Solhaug, I. E. (2014). Optimization of combined ship routing and inventory management in the salmon farming industry. Master's thesis, Norwegian Institute of Science and Technology, Trondheim, Norway.

Jaillet, P., Bard, J. F., Huang, L., and Dror, M. (2002). Delivery cost approximations for inventory routing problems in a rolling horizon framework. *Transportation Science*, 36(3):292–300.

Johansson, O. M. (2006). The effect of dynamic scheduling and routing in a solid waste management system. *Waste Management*, 26(8):875–885.

Kim, B. I., Kim, S., and Sahoo, S. (2006). Waste collection vehicle routing problem with time windows. *Computers & Operations Research*, 33(12):3624–3642.

Larson, R. C. (1988). Transporting sludge to the 106-mile site: An inventory/routing model for fleet sizing and logistics system design. *Transportation Science*, 22(3):186–198.

Mancini, S. (2015). A real-life multi depot multi period vehicle routing problem with a heterogeneous fleet: Formulation and adaptive large neighborhood search based matheuristic. *Transportation Research Part C: Emerging Technologies*.

Markov, I., de Lapparent, M., Bierlaire, M., and Varone, S. (2015). Modeling a waste disposal process via a discrete mixture of count data models. In *Proceedings of the 15th Swiss Transport Research Conference (STRC)*, April 17–19, 2015, Ascona, Switzerland.

Markov, I., Varone, S., and Bierlaire, M. (2016). Integrating a heterogeneous fixed fleet and a flexible assignment of destination depots in the waste collection VRP with intermediate facilities. *Transportation Research Part B: Methodological*, 84:256–273.

Mes, M., Schutten, M., and Rivera, A. P. (2014). Inventory routing for dynamic waste collection. *Waste Management*, 34(9):1564–1576.

Moghaddam, N. M. (2015). The partially rechargeable electric vehicle routing problem with time windows and capacitated charging stations. Master's thesis, Clemson University, Clemson, SC, USA.

Moin, N. H. and Salhi, S. (2007). Inventory routing problems: A logistical overview. *The Journal of the Operational Research Society*, 58(9):1185–1194.

Muter, I., Cordeau, J.-F., and Laporte, G. (2014). A branch-and-price algorithm for the multi-depot vehicle routing problem with interdepot routes. *Transportation Science*, 48(3):425–441.

Nekooghadirli, N., Tavakkoli-Moghaddam, R., Ghezavati, V., and Javanmard, S. (2014a). Solving a new bi-objective location-routing-inventory problem in a distribution network by meta-heuristics. *Computers & Industrial Engineering*, 76:204–221.

Nekooghadirli, N., Tavakkoli-Moghaddam, R., and Ghezavati, V. R. (2014b). Efficiency of a multi-objective imperialist competitive algorithm: A biobjective location-routing-inventory problem with probabilistic routes. *Journal of AI and Data Mining*, 2(2):105–112.

Nolz, P. C., Absi, N., and Feillet, D. (2014). A stochastic inventory routing problem for infectious medical waste collection. *Networks*, 63(1):82–95.

Park, Y.-B., Yoo, J.-S., and Park, H.-S. (2016). A genetic algorithm for the vendor-managed inventory routing problem with lost sales. *Expert Systems with Applications*, 53:149–159.

Pelletier, S., Jabali, O., and Laporte, G. (2014). Goods distribution with electric vehicles: Review and research perspectives. Technical Report CIRRELT-2014-44, CIRRELT, Montreal, Canada.

Penna, P. H. V., Subramanian, A., and Ochi, L. S. (2013). An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. *Journal of Heuristics*, 19(2):201–232.

Ribeiro, R. and Lourenço, H. (2003). Inventory-routing model for a multi-period problem with stochastic and deterministic demand. Technical Report 275, Department of Economics and Business, Universitat Pompeu Fabra, Barcelona, Spain.

Rochat, Y. and Taillard, É. D. (1995). Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics*, 1:147–167.

Ropke, S. and Pisinger, D. (2006a). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4):455–472.

Ropke, S. and Pisinger, D. (2006b). A unified heuristic for a large class of vehicle routing problems with backhauls. *European Journal of Operational Research*, 171(3):750–775.

Sassi, O., Cherif, W. R., and Oulamara, A. (2014). Vehicle routing problem with mixed fleet of conventional and heterogenous electric vehicles and time dependent charging costs, Working paper, University of Lorraine, Nancy, France.

Savelsbergh, M. W. P. (1992). The vehicle routing problem with time windows: Minimizing route duration. *INFORMS Journal on Computing*, 4(2):146–154.

Schneider, M., Stenger, A., and Goeke, D. (2014). The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science*, 48(4):500–520.

Schneider, M., Stenger, A., and Hof, J. (2015). An adaptive VNS algorithm for vehicle routing problems with intermediate stops. *OR Spectrum*, 37(2):353–387.

Shaw, P. (1997). A new local search algorithm providing high quality solutions to vehicle routing problems. Technical report, APES Group, Department of Computer Sciences, University of Strathclyde, Glasgow, Scotland.

Solyalı, O., Cordeau, J.-F., and Laporte, G. (2012). Robust inventory routing under demand uncertainty. *Transportation Science*, 46(3):327–340.

Soysal, M., Bloemhof-Ruwaard, J. M., Haijema, R., and van der Vorst, J. G. (2015). Modeling an inventory routing problem for perishable products with environmental considerations and demand uncertainty. *International Journal of Production Economics*, 164:118–133.

Soysal, M., Bloemhof-Ruwaard, J. M., Haijema, R., and van der Vorst, J. G. (2016). Modeling a green inventory routing problem for perishable products with horizontal collaboration. *Computers & Operations Research*, pages –.

Stewart, W. R. and Golden, B. L. (1983). Stochastic vehicle routing: A comprehensive approach. *European Journal of Operational Research*, 14(4):371–385.

Subramanian, A., Penna, P. H. V., Uchoa, E., and Ochi, L. S. (2012). A hybrid algorithm for the heterogeneous fleet vehicle routing problem. *European Journal of Operational Research*, 221(2):285–295.

Taillard, É. D. (1999). A heuristic column generation method for the heterogeneous fleet VRP. *RAIRO - Operations Research*, 33(1):1–14.

Tavares, G., Zsigraiova, Z., Semiao, V., and Carvalho, M. (2009). Optimisation of MSW collection routes for minimum fuel consumption using 3D GIS modelling. *Waste Management*, 29:1176–1185.

Trudeau, P. and Dror, M. (1992). Stochastic inventory routing: Route design with stockouts and route failures. *Transportation Science*, 26(3):171–184.

Yu, Y., Chu, C., Chen, H., and Chu, F. (2012). Large scale stochastic inventory routing problems with split delivery and service level constraints. *Annals of Operations Research*, 197(1):135–158.