Waste Collection Inventory Routing with Non-stationary Stochastic Demands

Iliya Markov[a,*], Michel Bierlaire[a], Jean-François Cordeau[b], Yousef Maknoon[c], Sacha Varone[d]

[a] *Transport and Mobility Laboratory*
School of Architecture, Civil and Environmental Engineering
École Polytechnique Fédérale de Lausanne
Station 18, 1015 Lausanne, Switzerland

[b] *CIRRELT and HEC Montréal*
3000 chemin de la Côte-Sainte-Catherine
Montréal, Canada H3T 2A7

[c] *Faculty of Technology, Policy, and Management*
Delft University of Technology
Jaffalaan 5, 2628 BX Delft, The Netherlands

[d] *Haute École de Gestion de Genève*
University of Applied Sciences Western Switzerland (HES-SO)
Campus Battelle, Rue de la Tambourine 17, 1227 Carouge, Switzerland

Abstract

We solve a rich routing problem inspired from practice, in which a heterogeneous fixed fleet is used for collecting recyclable waste from large containers over a finite planning horizon. Each container is equipped with a sensor that communicates its level at the start of the day. Given a history of observations, a forecasting model is used to estimate the expected demands and a forecasting error representing the level of uncertainty. The problem falls under the framework of the stochastic inventory routing problem and our main contribution is the modeling of the dynamic probability-based cost of container overflows and route failures over the planning horizon. We cast the problem as a mixed integer non-linear program and, to solve it, we develop an adaptive large neighborhood search algorithm that integrates a purpose-designed forecasting model, tested and validated on real data. We demonstrate the strength of our modeling approach on a set of rich inventory routing instances derived from real data coming from the canton of Geneva, Switzerland. Our approach significantly outperforms alternative deterministic policies in its ability to limit the occurrence of container overflows for the same routing cost. Finally, we show the benefit of a rolling horizon solution and derive empirical lower and upper bounds on its cost.

Keywords: transportation; stochastic inventory routing problem; waste collection; demand forecasting; overflows

1. Introduction

Waste collection is one of the most important logistical activities performed by any municipality, and also one of the most expensive, with collection costs alone accounting for more than 70% of waste management costs (Tavares et al., 2009). Recycling, on the other hand, can alleviate problems related to

*Corresponding author
Email addresses: iliya.markov@epfl.ch (Iliya Markov), michel.bierlaire@epfl.ch (Michel Bierlaire), jean-francois.cordeau@hec.ca (Jean-François Cordeau), m.y.maknoon@tudelft.nl (Yousef Maknoon), sacha.varone@hesge.ch (Sacha Varone)

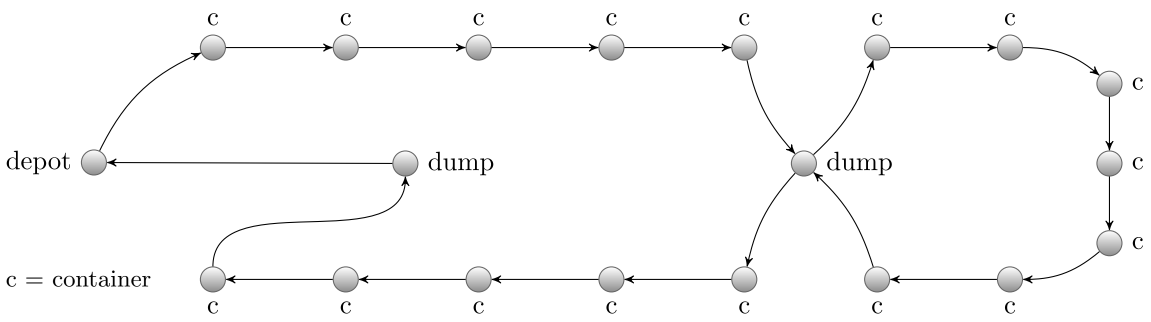
landfill capacity and pollution, and many countries have already set ambitious target levels for recycling. As part of its Circular Economy Strategy, the European Union (EU), for example, has adopted legislative proposals to set a common EU target for recycling 65% of municipal and 75% of packaging waste 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 the taxpayer.

In this context, we solve a rich recyclable waste collection problem, which can be described as follows. A heterogeneous fixed fleet is used for collecting recyclable waste from large containers. Each container holds and each vehicle collects a given waste flow, e.g. white glass, colored glass, paper, etc. Therefore, a separate problem is solved for each waste flow. As shown in Figure 1, a tour starts and ends at the depot, and is a sequence of collections followed by disposals at the available dumps. 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 containers, depots and dumps. For depots and dumps, time windows represent working hours, whereas for containers they can be set to avoid collections during rush hours, school times, etc. A tour is also limited by the legal duration of the working day. Accessibility restrictions apply to certain points, for example containers located in narrow streets that cannot be accessed by big collector trucks.

Containers are equipped with ultrasound sensors that communicate their waste levels via the GSM network at the start of the day. Yet, planning today's collections while ignoring future container demands is myopic. For instance, it may be cheaper to postpone a collection if we know that the container in question fills up slowly. On the other hand, it may be worthwhile collecting an almost empty container today if we know that it will experience high demand in the coming days. Since future demands are unknown, we use historical data and forecasting techniques to estimate their expected values and the distribution of the forecasting error term over a multi-day planning horizon.

Deviations of the demand realizations from their expected values may lead to undesirable events. If a container fills up more quickly than expected and is not collected on time, it may overflow. Experience suggests that such containers continue serving demand because people place the waste beside them. Nevertheless, the municipal regulations require that overflowing containers should be collected on the same day. By the same logic, if the containers planned for collection on a given day are fuller than

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



expected, the vehicle may run out of capacity before its scheduled dump visit, resulting in a so-called route failure. These undesirable events require recourse actions, such as the emergency collection of an overflowing container or an unplanned visit to a dump in the case of route failure. Recourse actions are expensive. Therefore, minimizing collection costs requires forward-looking plans with collection schedules that incorporate the probabilities of undesirable events.

Given the multi-day planning horizon and the uncertainty implied by the forecasting error, our problem falls under the framework of the Stochastic Inventory Routing Problem (SIRP). Markov et al. (2018) present a unified framework for rich routing problems with stochastic demands, which allows the explicit modeling of recourse actions, their probabilities and costs in a computationally tractable way. The present work builds on this unified framework by going much deeper into the specifics of our problem. We apply a forecasting model and derive the expressions for the recourse costs of container overflows and route failures. Containers additionally incur an overflow cost that the collector pays to the municipality in the form of a fine. The correct attribution of these costs 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 complicate the solution methodology by introducing non-linearities.

While the problem is stochastic, it is also dynamic with new container level information revealed daily. The latter is integrated in a rolling horizon fashion by solving an SIRP on each day and implementing the decisions for that day. Considering a multi-day planning horizon at each rollover thus takes advantage of future probabilistic information for making forward-looking decisions today. As a result, the collection tours executed on each day reflect the anticipation of future demands, balancing collection costs and the expected costs of undesirable events and their recourse actions. In our problem, demand is revealed in discrete time periods, i.e. with the start of each day. Conceptually, the availability of continuous demand information can be handled using a finer time discretization under the same modeling framework.

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 develops the mathematical formulation of our problem. Section 4 describes the solution methodology, followed by Section 5, which presents the numerical experiments. Finally, Section 6 concludes and explores future work directions.

2. Related Literature

Given the rich features of our problem, we review both the related VRP and IRP literature. In Section 2.1 below, we provide a short survey of the VRP with Intermediate Facilities (VRP-IF), the electric and alternative fuel VRP, and the heterogeneous fixed fleet VRP. Then, 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. Finally, in Section 2.3, we position our contribution.

2.1. Related VRP Literature

One of the seminal applications of the VRP to waste collection is that of Beltrami & Bodin (1974) who solve a periodic VRP-IF for commercial waste collection in New York City. Angelelli & Speranza

(2002) apply a tabu search heuristic to a similar periodic problem. Kim et al. (2006) develop a simulated annealing heuristic for a waste collection VRP-IF which also considers 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 decompose the problem into multi-depot, single-depot and inter-depot subproblems, which are solved by tabu search. A solution to the MDVRPI is obtained through a set covering formulation. 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. The best known heuristic solutions to these instances are due to Hemmelmayr et al. (2013) who develop a Variable Neighborhood Search (VNS). Muter et al. (2014) propose a branch-and-price algorithm for the MDVRPI and manage to solve to optimality 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 & Figliozzi (2011) consider the recharging VRP, where electric vehicles can recharge at customer locations with time windows. Erdoğan & 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 recent survey of the relevant literature is available in Pelletier et al. (2016).

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. 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 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 & Mingozzi (2009).

The vehicle routing component 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 by 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. As motivated in Section 1, 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, meaning that

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
Inventory decisions	Back-ordering (with a penalty and limit)
Fleet composition	Heterogeneous
Fleet size	Multiple (fixed)

a visited container is always fully emptied. Container overflow is served at a penalty, which implies back-ordering inventory decisions where the number of back-order days is limited to one. The fleet is heterogeneous and fixed. Information-wise, the problem is stochastic and, given that it is solved in a rolling horizon fashion, dynamic with new container information revealed each day. Other comprehensive surveys with a particular focus on the stochastic IRP can be found in Moin & Salhi (2007) and Yu et al. (2012). 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 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 & Dror (1992) extend the work of Dror & Ball (1987) on the optimal service frequency under a stochastic setting. They consider both stock-outs and route failures. Unlike previous research which uses a vehicle with an artificially small capacity to avoid route failures, Trudeau & Dror (1992) develop an analytical probability expression, and corroborate their modeling approach with a simulation experiment. Our work differs from that of Trudeau & Dror (1992) in several major aspects. First, we have a heterogeneous fixed fleet rather than a homogeneous one. Second, route failures pertain to sections of the tours called trips, which are delimited by dump visits where the vehicle capacity is renewed. Thus, there can be several trips and hence several route failures in any 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 tracking of each container's visit-dependent and conditional probability of overflow on each day of the planning horizon. On top of that, we consider multiple rich routing features.

The work of Bard et al. (1998) 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 shifted 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. 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 & 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 & Ball, 1987; Trudeau & Dror, 1992; Dror & 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 & 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. A robust approach is also used by Roldán et al. (2016) and Rahbari et al. (2017). 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. (2018).

2.3. Discussion

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 orientation. However, it still leaves open 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. In a rolling horizon framework, 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 undesirable events, and this approach has a clear cost orientation, 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. Our IRP formulation is cost-oriented and includes 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.

Markov et al. (2018) propose a framework for modeling demand stochasticity using a cost-oriented approach, which unifies a number of routing problems from various application fields, including health care, maritime operations, waste collection and facility maintenance. The authors focus on topics that address existing gaps between the literature and practice. They demonstrate the negligible effect that the used modeling simplifications have on the objective function and show that stochasticity can be handled just as effectively in the objective and in the constraints. Although the framework has widespread applicability, several of its key components need to be tailored for the specific problem at hand. The purpose of the present paper is to explain how the framework can be turned into an operational model for the specific case of the waste collection problem.

In this context, we solve a waste collection problem that is broad enough to fit many practical situations. The contribution of the present work is three-fold. First, we derive the probabilities and develop the expression for the dynamic probability-based costs of container overflows and route failures over the planning horizon. Second, we propose a state-of-the-art Adaptive Large Neighborhood Search (ALNS) algorithm and integrate it with a purpose-designed forecasting model, which has been tested and validated on waste collection data. The algorithm is able to handle a variety of rich routing features traditionally absent or rarely considered in the IRP literature, such as a heterogeneous fixed fleet and intermediate facilities. Third, we demonstrate the strength of our modeling approach on a set of rich IRP instances derived from real data coming from the canton of Geneva, Switzerland. Our approach performs significantly better than alternative deterministic policies in its ability to control the occurrence of container overflows for the same routing cost. We show the benefit of the rolling horizon approach that includes the newly revealed container information each day and derive empirical lower and upper bounds on its cost.

3. Formulation

In what follows, Section 3.1 presents a brief sketch of the forecasting model and Section 3.2 develops the mathematical formulation for our SIRP. Table 2 summarizes the notations used. We highlight 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 waste flow, one of them becomes limiting while the other may not be. However, 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

Table 2: Notations

Sets			
$\mathcal{V}$	set of distinct 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 $= \{0, \ldots, u\}$	$\mathcal{T}^+$	shifted planning horizon $= \{1, \ldots, u, u+1\}$
$\mathscr{S}_{kt}$	set of depot-to-dump or dump-to-dump trips for vehicle $k \in \mathcal{K}$ on day $t \in \mathcal{T}$	$\mathcal{S}$	set of containers in a particular trip in $\mathscr{S}_{kt}$

Parameters	
ξ_{itg}	Poisson rate for deposit volume v of container i on day t
$\boldsymbol{\kappa}_{it}$	vector of covariates for container i on day t
$\boldsymbol{\gamma}_v$	vector of estimable parameters for deposit volume v
ρ_{it}	demand of container i on day t (random variable)
ε_{it}	error term of container i on day t
ς	forecasting 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 $\mathcal{S} \in \mathscr{S}_{kt}$ to the nearest dump and back to $\mathcal{S}$ (monetary)

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)

step, we avoid tracking both volume and weight for the benefit of a more elegant formulation.

3.1. Forecasting Model

Any model can be applied to forecast the expected container demands over the planning horizon and to derive the distribution of the forecasting error (Markov et al., 2018). Here, we use the model proposed by Markov et al. (2015), 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 though simplified underlying behavior. We assume a set $\mathcal{V}$ of distinct deposit volumes, where deposit volume $v \in \mathcal{V}$ is generated with a Poisson rate ξ_{itv} for container i on day t. The rate ξ_{itv} takes the functional form $\xi_{itv} = \exp(\boldsymbol{\kappa}_{it}^{\top}\boldsymbol{\gamma}_v)$, where $\boldsymbol{\kappa}_{it}$ is a vector of covariates, such as the day of the week, weather variables, holiday periods, etc., and $\boldsymbol{\gamma}_v$ is a vector of estimable parameters for deposit volume v. We formulate an expression for the expected value of the demand of container i on day t as follows:

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

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 availability:

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

assuming strict exogeneity and with error terms represented by white noise as

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

and where a consistent estimate of the variance is given by

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

We refer to ς as the forecasting error. 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 $\mathcal{T} = \{0, \ldots, u\}$ and 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 by 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.

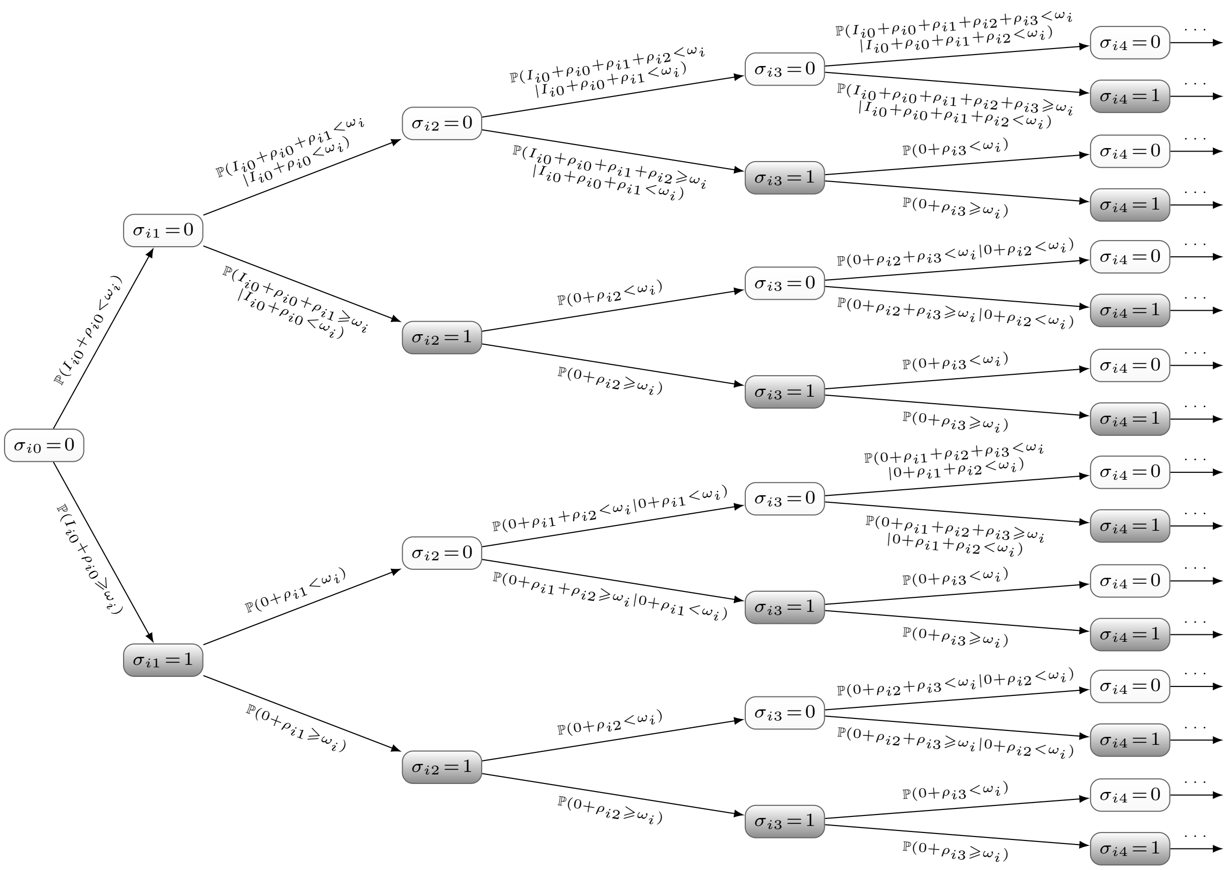
3.2.1. Derivation of the Overflow Probabilities.

In line with Markov et al.'s (2018) framework, 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 & Ball, 1987; Trudeau & 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 pays to the municipality 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.

According to Markov et al. (2018), the overflow probability evolution can be represented by a succession of binary trees which can be precomputed for all possible collection scenarios. Assume that we start with an inventory I_{i0} such that container i is initially in state $\sigma_{i0} = 0$. With no collections during the planning horizon, the state probability tree develops as in Figure 2. 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 as the inventory is reset to zero by the emergency collection. We are only interested in the probability of overflow, i.e. of being in a state $\sigma_{it} = 1$, which 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 $t = 2$. For container i, the exhaustive list of probabilities to precompute 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 $\mathbb{P}(0 + \sum_{t=g}^{h} \rho_{it} \geqslant \omega_i \mid 0 + \sum_{t=g}^{h-1} \rho_{it} < \omega_i), \forall g, h \in \mathcal{T} \colon h > g$.

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



The calculation of the conditional probabilities involves the evaluation of:

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

As explained in Section 3.1, any forecasting model can be used as long as it provides the distribution of the error terms. This distribution is used in calculating the overflow probabilities. In our case, given 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 $X\sim\mathcal{N}(0,(h-g)\varsigma^2)$ and X is independent of ε_{ih}. Formula (6) then rewrites as:

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

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

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

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.

Using the above derivations, we formulate the objective function z in line with the structure of Markov et al.'s (2018) unified framework. It consists of the Expected Overflow and Emergency Collection Cost (EOECC), the Routing Cost (RC), and the Expected Route Failure Cost (ERFC). The objective function (9) is non-linear due to the non-linear nature of the EOECC and the ERFC components defined below:

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

The expected overflow and emergency collection cost is expressed as:

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

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 the container has not undergone any regular collections before day t. The state probability is calculated by multiplication of the involved branch probabilities described in Section 3.2.1. 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, the EOECC is computed for $t \in \mathcal{T} \cup \mathcal{T}^+$, where $\mathcal{T}^+ = \{1, \ldots, u, u+1\}$.

The routing cost reflects the daily deployment cost, the distance-related cost and the time-related cost for each vehicle used over the planning horizon. It is formulated as:

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

The expected route failure cost reflects the vehicles' inability to serve the containers due to insufficient capacity on the scheduled depot-to-dump or a dump-to-dump trips. It is expressed as:

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

where $\mathscr{S}_{kt}$ 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 set $\mathscr{S}_{kt}$ is generated by inspecting the routing variables x_{ijkt}. At every feasible solution, for each vehicle k on each day t we can inspect the point visit sequence encoded in the variables x_{ijkt} to generate the set of depot-to-dump and dump-to-dump trips. 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 collection before day t. Given the order-up-to level inventory policy and a container s:

$$\Lambda_{sm} = \begin{cases} 0 & \text{if } \exists k \in \mathcal{K} : y_{skm} = 1, \\ I_{s0} & \text{if } y_{sk0} = 0, \forall k \in \mathcal{K}. \end{cases} \tag{13}$$

In other words, $\Lambda_{sm} = 0$ if there is a regular collection of container s on day m, and is equal to the initial inventory I_{s0} if there is no regular collection on day 0. In essence, 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.

The nearest dump to each container can be precomputed. Probability-wise, once the days m 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:

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

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 & Stegun, 1972). These repetitive calculations have no discernible influence on the algorithm's runtime.

As shown in Markov et al. (2018), the expressions of the RC and ERFC components imply a slight overestimation of the real cost as they ignore the probability of containers overflowing before the day t on which they are collected. If such containers are skipped in the tours performed on day t, this would reduce the RC. Whether they are skipped or not, they would be less full than expected, which would lower the ERFC due to the lower probability of the collected volume on day t exceeding the vehicle capacity. Trudeau & Dror (1992) develop probabilistic expressions that capture these effects for a simpler setup and with various assumptions that we do not impose, including a single container visit and overflow over the planning horizon. On the other hand, given the operational nature of the problem, implementing a full-blown simulator of the objective function to capture these effects would be very impractical. Still, as demonstrated by the numerical experiments in Section 5, after simulation the number of realized overflows over the planning horizon is so low that these two un-captured effects are marginal. Therefore, we consider our model a good representation of reality.

3.2.3. Constraints.

The constraints can be split into several categories, with the first category consisting of basic vehicle routing constraints. Constraints (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. Constraints (19) guarantee that vehicles do not visit inaccessible points. Flow conservation is represented by constraints (20).

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

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

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

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

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

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

The inventory constraints are necessary for tracking the container inventories and linking them to the vehicle visits and the pickup quantities. Constraints (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 3.2.2, 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. Constraints (24) force the pickup quantity to zero if the container is not visited. Constraints (25) and (26) represent the order-up-to 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 container 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, constraints (27) bound from below the cumulative quantity on the vehicle at each container, while constraints (28) enforce the vehicle capacity. Constraints (29) reset the cumulative quantity on the vehicle to zero at the origin, destination, and dumps. Keeping track of the cumulative quantity on the vehicle is achieved by constraints (30).

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

$$Q_{ikt} \leqslant \Omega_k, \qquad \forall t \in \mathcal{T}, k \in \mathcal{K}, i \in \mathcal{P} \tag{28}$$

$$Q_{ikt} = 0, \qquad \forall t \in \mathcal{T}, k \in \mathcal{K}, i \in \mathcal{N} \setminus \mathcal{P} \tag{29}$$

$$Q_{ikt} + q_{jkt} \leqslant Q_{jkt} + \Omega_k \left(1 - x_{ijkt}\right), \qquad \forall t \in \mathcal{T}, k \in \mathcal{K}, i \in \mathcal{N} \setminus \{d\}, j \in \mathcal{P} \tag{30}$$

The next four constraints express the intra-day temporal characteristics of the problem. Constraints (31) 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 (32) and (33) enforce the time windows. Constraints (34) provide a lower bound on the tour duration, while constraints (35) apply the maximum tour duration.

$$S_{ikt} + \delta_i + \tau_{ijk} \leqslant S_{jkt} + (\mu_i + \delta_i + \tau_{ijk})(1 - x_{ijkt}), \quad \forall t \in \mathcal{T}, k \in \mathcal{K}, i \in \mathcal{N} \setminus \{d\}, j \in \mathcal{N} \setminus \{o\} \tag{31}$$

$$\lambda_i \sum_{j \in \mathcal{N}} x_{ijkt} \leqslant S_{ikt}, \quad \forall t \in \mathcal{T}, k \in \mathcal{K}, i \in \mathcal{N} \setminus \{d\} \tag{32}$$

$$S_{jkt} \leqslant \mu_j \sum_{i \in \mathcal{N}} x_{ijkt}, \quad \forall t \in \mathcal{T}, k \in \mathcal{K}, j \in \mathcal{N} \setminus \{o\} \tag{33}$$

$$S_{dkt} - S_{okt} \geqslant 0, \quad \forall t \in \mathcal{T}, k \in \mathcal{K} \tag{34}$$

$$S_{dkt} - S_{okt} \leqslant \mathrm{H}, \quad \forall t \in \mathcal{T}, k \in \mathcal{K} \tag{35}$$

Finally, constraints (36) and (37) establish the variable domains.

$$x_{ijkt}, y_{ikt}, z_{kt} \in \{0, 1\}, \quad \forall t \in \mathcal{T}, k \in \mathcal{K}, i, j \in \mathcal{N} \tag{36}$$

$$q_{ikt}, Q_{ikt}, I_{it}, S_{ikt} \geqslant 0, \quad \forall t \in \mathcal{T}, k \in \mathcal{K}, i \in \mathcal{N} \tag{37}$$

4. Adaptive Large Neighborhood Search

Adaptive Large Neighborhood Search (ALNS) was introduced by Ropke & 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:

$$W_i = \begin{cases} W_i & \text{if } N_i^F = 0, \\ (1-b)W_i + bC_i^F / \left(m_i N_i^F\right) & \text{otherwise.} \end{cases} \tag{38}$$

In expression (38), m_i is a normalization factor damping the weights of more computationally expensive operators by multiplying the number of times they were applied (Ropke & Pisinger, 2006b; Coelho et al., 2012). 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.

Regarding the initial solution, 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.

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 instances. 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:

$$V^{\Omega}(s) = \sum_{t\in\mathcal{T}} \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{P}'_{kt}} (Q_{ikt} - \Omega_k)^+. \tag{39}$$

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:

$$V^{\mu}(s) = \sum_{t\in\mathcal{T}} \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{N}'_{kt}} (S_{ikt} - \mu_i)^+. \tag{40}$$

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:

$$V^{\mathrm{H}}(s) = \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} (S_{dkt} - S_{okt} - \mathrm{H})^{+}. \tag{41}$$

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

$$V^{\omega}(s) = \sum_{t \in \mathcal{T}^{+}} \sum_{i \in \mathcal{P}} (I_{it} - \omega_i)^{+}. \tag{42}$$

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:

$$V^{0}(s) = \sum_{i \in \mathcal{P}} \left(\left(1 - \sum_{k \in \mathcal{K}} y_{ik0} \right) (I_{i0} - \omega_i)^{+} \right). \tag{43}$$

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:

$$V^{\alpha}(s) = \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{N}'_{kt}} (y_{ikt} - \alpha_{ik})^{+}. \tag{44}$$

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

$$f(s) = z(s) + L^{\Omega} V^{\Omega}(s) + L^{\mu} V^{\mu}(s) + L^{\mathrm{H}} V^{\mathrm{H}}(s) + L^{\omega} V^{\omega}(s) + L^{0} V^{0}(s) + L^{\alpha} V^{\alpha}(s). \tag{45}$$

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 multiplied by a rate $\ell > 1$, otherwise it is divided by the same rate. If s has no feasibility violation, the values of $f(s)$ and $z(s)$ coincide.

4.2. Operators

The main ingredient of the ALNS are the destroy and repair operators. Some of the operators are inspired or adapted from the literature (Ropke & Pisinger, 2006a,b; Coelho et al., 2012; Buhrkal et al., 2012; Hemmelmayr et al., 2013), while others are developed to capture the specifics of our problem, in particular the stochastic objective function and the presence of 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 $\Delta f^{\max}$ in the solution cost. It is applied ν times.
3. *Shaw removals with relatedness.* Based on the ideas of Shaw (1997) and Ropke & Pisinger (2006a), this operator removes containers based on a relatedness measure among them. It starts by selecting a random tour and a random container i in this tour, and computing the relatedness R_{ij} of i to all containers j in the tours scheduled on the same day t as the randomly selected tour. We define the relatedness measure R_{ij} of container i to container j as:

$$R_{ij} = d_1 \pi_{ij}^{[0,1]} + d_2(|\lambda_i - \lambda_j| + |\mu_i - \mu_j|)^{[0,1]} + d_3|o_{it} - o_{jt}|^{[0,1]}, \quad (46)$$

where the first term captures the distance, the second terms captures the time window difference and the third term captures the overflow probability difference on day t. As in expression (10) for the EOECC, the latter is given by:

$$o_{it} = \mathbb{P}\left(\sigma_{it} = 1 \mid m = \max\left(0, g < t \colon \exists k \in \mathcal{K} \colon y_{ikg} = 1\right)\right). \quad (47)$$

These terms are scaled between zero and one, as indicated in superscript, and weighted by the parameters d_1, d_2 and d_3. The relatedness measures are again scaled between zero and one. Container i and all containers j for which R_{ij} is less than a threshold d_4 are removed.
4. *Remove container cluster:* Inspired by the work of Ropke & Pisinger (2006b), this operator removes large clusters of containers. It selects a random day t in the planning horizon and divides the containers visited on this day into k clusters, where k is chosen to be the number of tours executed on this day. If there is only one tour, its containers are divided into 2 clusters. Clustering is performed using Kruskal's algorithm, which progressively merges the containers into clusters based on distance, until the required number k of clusters is reached. Finally, a cluster is chosen randomly and removed as long as it contains less than half of the containers visited on day t.
5. *Empty a random day.* This operator selects a random day and empties all tours performed on it.
6. *Empty a random vehicle.* This operator selects a random vehicle and empties the tours performed by it on all days.
7. *Remove a random dump.* This operator selects a random tour and a random dump in it, excluding the last dump, and removes it.
8. *Remove the worst dump.* This operator removes the dump that would lead to the largest savings $\Delta f^{\max}$ in the solution cost. The last dump in each tour is never removed.
9. *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 visit the same container 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 minimum increase $\Delta f^{\min}$ in the solution cost. It is applied ν times.

2. *Insert ν containers in the best way.* This operator identifies for each container $i \in \mathcal{P}$ the tour and the position in that tour that would lead to the minimum increase $\Delta f_i^{\min}$ in the solution cost if the container is inserted there, checking that the container is not visited on the day the tour is performed. The containers in $\mathcal{P}$ are sorted in ascending order of $\Delta f_i^{\min}$ and the first ν of them are inserted in the previously identified tours and positions.

3. *Insert ν containers with regret-k:* As noted in Ropke & Pisinger (2006a), the motivation for using regret is to introduce a look-ahead information in the insertion process. Let Y_{ik} indicate the tour in which inserting container i using best insertion leads to the k^{th} lowest increase in the solution cost $\Delta f_{i,Y_{ik}}$. For a container i, we define the regret-k value as $c_i^k = \Delta f_{i,Y_{ik}} - \Delta f_{i,Y_{i1}}$, i.e. the difference between inserting the container in its best tour and its k^{th} best tour. It may be impossible to insert some containers in k different tours, thus the regret is computed for the largest possible $k' \leqslant k$. The containers in $\mathcal{P}$ are sorted in ascending order of k' and descending order of $c_i^{k'}$. The first ν containers in the ordered list are inserted in the tours and positions that would lead to the minimum increase $\Delta f^{\min}$ in the solution cost. In other words, we insert the containers that we will regret the most if they are not inserted now.

4. *Shaw insertions with relatedness.* This operator selects a random day t and a random container $i \in \mathcal{P}$ not visited on day t. It then proceeds to find the relatedness measure R_{ij}, as defined by formula (46), to all containers $j \in \mathcal{P}$ also not visited on day t. It inserts the container i as well as all containers j not visited on day t, for which R_{ij} is lower than a threshold d_4, in the tours executed on day t and in the positions that would lead to the minimum increase $\Delta f^{\min}$ in the solution cost.

5. *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.

6. *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.

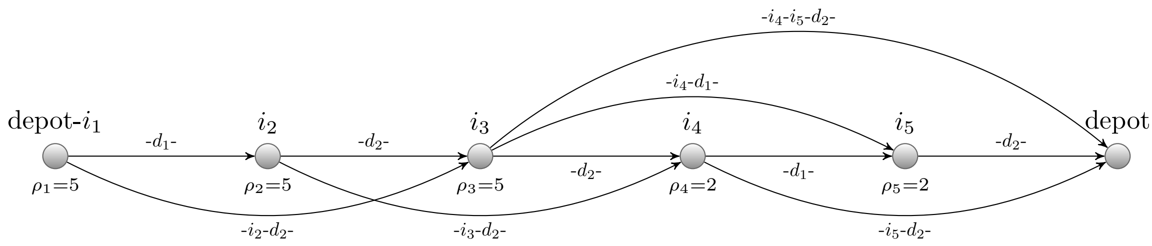
7. *Insert a dump in the best way.* This operator selects a random dump from $\mathcal{D}$ and inserts it in the tour and in the position that would lead to the minimum increase $\Delta f^{\min}$ in the solution cost.

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

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

10. *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.

Figure 3: Feasibility Graph of the Reorder Dumps Operator



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

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 16.04.2. Each instance is solved 10 times, out of which we report the best, average and worst result, or averaged values of the best, average and worst result over a set of instances, unless indicated otherwise. Section 5.1 below explains how the algorithmic parameters were tuned. Section 5.2 describes the case study instances and Section 5.3 presents an extensive analysis of the stochastic modeling approach, its comparison to alternative deterministic policies, and its application in a rolling horizon framework. The Online Appendices provide additional experiments evaluating the quality of the ALNS with tests on classical IRP and VRP benchmark instances, some of which represent typical waste collection configurations.

5.1. Parameter Tuning

The algorithmic parameters were tuned on the classical Archetti et al. (2007) IRP benchmark instances as well as on the case study instances described in Section 5.2 next. We first tuned the SA-related parameters followed by the ALNS-related and the operator-related parameters. Initial values were either

Table 3: Algorithmic Parameters

SA-Related		ALNS-Related		Operator-Related	
Parameter	Value	Parameter	Value	Parameter	Value
Initial temperature (T^{start})	10,000	F segment length	2000	Rel. weight d_1	0.54
Start temp. control param. (w)	0.6	Reaction factor (b)	0.5	Rel. weight d_2	0.23
Cooling rate (r)	0.99998	Reward e_1	30	Rel. weight d_3	0.23
Final temperature (T^{end})	0.01	Reward e_2	20	Rel. threshold d_4	0.2/0.3
Penalty change rate (ℓ)	1.06	Reward e_3	5	Regret-k	2

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. Then 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 strategy is to limit the search at very high temperatures (Ropke & 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 (39) through (44) 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. The two destroy operators *Shaw removals with relatedness* and *remove container cluster* were given normalization factors m_i of 8, and the two repair operators *insert ν containers in the best way* and *insert ν containers with regret-k* were given normalization factors m_i of 4.5. The normalization factors for the rest are all equal to one. For the operator-related parameters, the best results were obtained for regret-2. The relatedness weights d_1, d_2 and d_3 were calibrated at 0.54, 0.23 and 0.23, respectively. The relatedness thresholds of 0.2 for removals and 0.3 for insertions performed the best.

5.2. Case Study Instances

The case study data includes 63 instances of white glass collections in the canton of Geneva, Switzerland, created using the historical records for the years 2014, 2015 and 2016. Each instance has a weekly planning horizon 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. 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. The fleet consists, depending on the instance, of one or 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. We also have access to historical waste levels for each container. Thus, the demands for each instance are forecast by the model from Section 3.1 using the previous 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 ς estimated by formula (4). We do not have information about maximum tour duration, time windows and the cost parameters, for which we set realistic or reasonable values. Thus, tours should respect a maximum duration of four hours each, and the time windows correspond to 8:00 a.m. until noon. 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.

5.3. Analysis of the Stochastic Approach

In this section, we analyze the proposed stochastic approach on the waste collection IRP instances introduced above. Section 5.3.1 evaluates the effect of incorporating the probability-based cost of container overflows and route failures in the objective function in terms of its impact on the total cost and the resulting frequency of these undesirable events. Section 5.3.2 compares the stochastic approach to alternative deterministic policies such as artificial buffer capacities for the containers and trucks. In both Sections 5.3.1 and 5.3.2, simulation of the stochastic demands is used to assess the quality of the produced solution. Finally, Section 5.3.3 presents the rolling horizon experiments and derives empirical lower and upper bounds on the solution cost for the planning horizon.

5.3.1. Probabilistic Policies.

To study the impact of the probability information included in the objective function, we perform two types of experiments on the instances described in Section 5.2. The first type considers the complete objective function with all relevant costs, as defined by expression (9). We label the problem with this objective “Complete”. The second type 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”. Since the routing-only problem ignores all stochastic information and only the stochastic information, it becomes the deterministic version of the stochastic problem. Tables 4, 5 and 6 summarize the numerical results for various choices of the Emergency Collection Cost (ECC) and the Route Failure Cost Multiplier (RFCM), and each row represents averaged values over the 63 instances. In these three tables, the first three columns identify the type of objective considered (complete vs. routing-only), the applied ECC, and the applied RFCM. In Table 4, the next four columns report the computation time, the average number of tours, container collections and dump visits. As each instance is solved 10 times, the next three columns report the average over the 63 instances of the best, average and worst cost, respectively, over the 10 runs for each instance. The last two columns show the percent gap between the average and the best cost, and the worst and the best cost, respectively. 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. Over the 63 instances, the average gap between the average cost and the best cost is in the order of 0.5%, and between the worst cost and the best cost it is in the order of 1.5%, which is an indication of the quality of the results provided by the ALNS. The values are lowest for the routing-only objective and grow with higher ECC for the complete objective, reflecting the more challenging search

Table 4: Probabilistic Policies: Basic Results for Cost Analysis on Real Data Instances

Objective	ECC	RFCM	Runtime(s.)	Avg Num Tours	Avg Num Collections	Avg Num Dump Visits	Best Cost (CHF)	Avg Cost (CHF)	Worst Cost (CHF)	Gap Avg-Best(%)	Gap Worst-Best(%)
Complete	100	1.00	870.65	1.95	44.41	2.24	662.65	666.64	672.87	0.60	1.54
Complete	100	0.50	871.84	1.95	44.45	2.25	662.38	666.57	673.30	0.63	1.65
Complete	100	0.25	885.52	1.95	44.46	2.24	662.38	666.92	673.15	0.69	1.63
Complete	100	0.00	871.81	1.95	44.46	2.23	662.26	666.78	674.01	0.68	1.78
Complete	50	1.00	864.57	1.92	42.39	2.18	648.14	651.36	656.77	0.50	1.33
Complete	50	0.50	855.51	1.92	42.40	2.17	647.99	651.50	656.90	0.54	1.37
Complete	50	0.25	873.28	1.92	42.36	2.16	648.05	651.15	656.66	0.48	1.33
Complete	50	0.00	856.39	1.92	42.35	2.18	648.14	651.40	656.47	0.50	1.29
Complete	25	1.00	841.94	1.90	41.03	2.16	638.61	641.41	646.06	0.44	1.17
Complete	25	0.50	844.22	1.90	41.05	2.16	638.38	641.22	645.89	0.44	1.18
Complete	25	0.25	846.67	1.90	41.01	2.15	638.57	641.50	646.19	0.46	1.19
Complete	25	0.00	855.83	1.90	41.01	2.15	638.42	641.49	646.36	0.48	1.24
Routing-only	0	0.00	681.27	1.83	16.64	1.87	421.99	422.48	423.12	0.12	0.27

space produced by the non-linearities present there. On the other hand, it appears that the gaps are almost unaffected by the RFCM.

Table 5 is a more detailed breakdown of the cost and efficiency structure of the set of objective functions presented in Table 4. The fourth, fifth and sixth columns decompose the average solution cost from Table 4 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 per unit of 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 40% 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 remains almost unchanged. A higher emergency collection cost necessitates more frequent visits as an attempt to further limit overflows. 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 20% higher with respect to the total cost, and almost 40% 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 for it. First, we sample the forecasting error independently for each container and each day using the estimate ς. Then, we evaluate the effect on the occurrence of container overflows and route

Table 5: Probabilistic 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	1.00	579.75	86.86	0.03	47,821.12	71.73	82.49
Complete	100	0.50	579.84	86.65	0.07	47,920.02	71.89	82.64
Complete	100	0.25	580.16	86.71	0.04	47,925.52	71.86	82.61
Complete	100	0.00	579.93	86.85	0.00	47,872.93	71.80	82.55
Complete	50	1.00	563.52	87.83	0.01	46,247.51	71.00	82.07
Complete	50	0.50	563.03	88.40	0.08	46,327.89	71.11	82.28
Complete	50	0.25	562.19	88.91	0.05	46,380.87	71.23	82.50
Complete	50	0.00	563.34	88.06	0.00	46,404.74	71.24	82.37
Complete	25	1.00	553.80	87.59	0.02	45,215.18	70.49	81.64
Complete	25	0.50	553.74	87.42	0.07	45,279.65	70.62	81.77
Complete	25	0.25	553.77	87.68	0.06	45,281.71	70.59	81.77
Complete	25	0.00	553.53	87.96	0.00	45,347.30	70.69	81.92
Routing-only	0	0.00	422.48	0.00	0.00	24,955.14	59.07	59.07

Table 6: Probabilistic 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	1.00	0.83	1.60	2.15	3.26	0.05	0.05	0.05	0.07
Complete	100	0.50	0.81	1.58	2.14	3.27	0.05	0.06	0.07	0.10
Complete	100	0.25	0.81	1.59	2.15	3.26	0.05	0.07	0.07	0.11
Complete	100	0.00	0.83	1.57	2.16	3.28	0.10	0.11	0.12	0.16
Complete	50	1.00	1.04	1.87	2.48	3.72	0.05	0.05	0.05	0.05
Complete	50	0.50	1.04	1.86	2.48	3.73	0.05	0.07	0.07	0.07
Complete	50	0.25	1.06	1.88	2.50	3.72	0.06	0.09	0.09	0.10
Complete	50	0.00	1.06	1.87	2.48	3.72	0.09	0.11	0.11	0.13
Complete	25	1.00	1.26	2.12	2.73	4.08	0.06	0.06	0.06	0.06
Complete	25	0.50	1.25	2.10	2.73	4.07	0.05	0.07	0.07	0.07
Complete	25	0.25	1.25	2.11	2.74	4.09	0.05	0.08	0.08	0.09
Complete	25	0.00	1.25	2.11	2.77	4.09	0.09	0.10	0.11	0.11
Routing-only	0	0.00	16.93	20.45	22.55	26.71	0.04	0.05	0.05	0.05

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 6 summarizes the number of overflows and route failures at the 75th, 90th, 95th and 99th percentiles of the 10,000 simulation runs for each instance, where each row is an averaged result for 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 the RFCM. What is more striking, however, is the difference between the series of complete objectives on the one hand and the routing-only objective on the other. While the complete objectives are able to limit the number of overflows to about four, even at the extreme of the simulated distribution, the average number of overflows when using the routing-only objective is higher by a degree of magnitude.

5.3.2. Alternative Policies.

To further study the theoretical justification and practical relevance of the stochastic modeling 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 policies and the alternative policies of artificially low capacities.

Tables 7, 8 and 9 are structured in the same way as Tables 4, 5 and 6 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 7, we note the strong negative correlation between the container effective capacity and the average number of tours, container collections and dump visits in the solutions. We also notice that the relative increase in the number of container collections 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

Table 7: Alternative Policies: Basic Results for Cost Analysis on Real Data Instances

Objective	CEC	TEC	Runtime(s.)	Avg Num Tours	Avg Num Collections	Avg Num Dump Visits	Best Cost (CHF)	Avg Cost (CHF)	Worst Cost (CHF)	Gap Avg-Best(%)	Gap Worst-Best(%)
Routing-only	1.00	1.00	682.31	1.83	16.64	1.87	421.95	422.51	423.16	0.13	0.29
Routing-only	1.00	0.90	685.38	1.83	16.65	1.87	422.22	422.80	423.47	0.14	0.30
Routing-only	1.00	0.75	672.96	1.83	16.65	1.95	423.38	424.02	424.92	0.15	0.36
Routing-only	1.00	0.60	757.33	1.83	16.66	2.04	425.31	426.06	426.93	0.18	0.38
Routing-only	0.90	1.00	742.70	2.00	22.63	2.02	486.29	486.83	487.59	0.11	0.27
Routing-only	0.90	0.90	746.77	2.00	22.62	2.06	486.82	487.39	488.09	0.12	0.26
Routing-only	0.90	0.75	738.18	2.00	22.62	2.15	488.46	489.16	489.95	0.14	0.31
Routing-only	0.90	0.60	725.43	2.00	22.63	2.37	492.74	493.71	494.69	0.20	0.39
Routing-only	0.75	1.00	873.54	2.00	33.52	2.43	541.87	542.92	544.53	0.19	0.49
Routing-only	0.75	0.90	863.36	2.00	33.52	2.60	544.60	545.78	547.25	0.22	0.49
Routing-only	0.75	0.75	869.94	2.00	33.50	2.86	549.13	550.15	551.46	0.19	0.42
Routing-only	0.75	0.60	862.67	2.00	33.54	3.12	555.35	557.37	559.75	0.36	0.79
Routing-only	0.60	1.00	1037.72	2.97	44.59	3.78	780.40	783.05	788.46	0.34	1.03
Routing-only	0.60	0.90	1241.91	2.97	44.65	3.88	782.50	785.42	792.24	0.37	1.25
Routing-only	0.60	0.75	1060.95	2.97	44.67	4.10	788.74	792.06	798.07	0.42	1.18
Routing-only	0.60	0.60	1023.95	2.97	44.79	4.58	799.71	804.37	811.70	0.58	1.50

long run. We notice that the solution time grows with the number of container collections, and so do the solution gaps. Yet, the increase of the solution time is smaller than the increase of the number of container collections. Moreover, even the highest gaps for a container effective capacity of 60% remain in the order of 1.5% and below. One explanation for the increase of solution time and the gaps could be that the problem becomes tighter and hence the solution space more challenging. In fact, two of the instances for a CEC of 60% are infeasible.

Table 8 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 reveal an interesting result. Lowering the CEC from 100%, to 90%, to 75% results in solutions collecting more volume, but also more volume per unit routing cost. However, further lowering the container effective capacity to 60% results in a disproportionately higher routing cost. As a result, despite collecting more volume, the solutions with a CEC of 60% are less efficient in terms of collected volume per unit routing cost compared to the solutions with a CEC of 75%. Table 9 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

Table 8: Alternative 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	422.51	0.00	0.00	24,992.02	59.15	59.15
Routing-only	1.00	0.90	422.80	0.00	0.00	24,963.64	59.04	59.04
Routing-only	1.00	0.75	424.02	0.00	0.00	24,986.17	58.93	58.93
Routing-only	1.00	0.60	426.06	0.00	0.00	24,909.59	58.46	58.46
Routing-only	0.90	1.00	486.83	0.00	0.00	31,553.37	64.81	64.81
Routing-only	0.90	0.90	487.39	0.00	0.00	31,577.74	64.79	64.79
Routing-only	0.90	0.75	489.16	0.00	0.00	31,747.19	64.90	64.90
Routing-only	0.90	0.60	493.71	0.00	0.00	31,846.97	64.51	64.51
Routing-only	0.75	1.00	542.92	0.00	0.00	44,149.46	81.32	81.32
Routing-only	0.75	0.90	545.78	0.00	0.00	44,108.02	80.82	80.82
Routing-only	0.75	0.75	550.15	0.00	0.00	43,985.69	79.95	79.95
Routing-only	0.75	0.60	557.37	0.00	0.00	44,219.61	79.34	79.34
Routing-only	0.60	1.00	783.05	0.00	0.00	54,332.98	69.39	69.39
Routing-only	0.60	0.90	785.42	0.00	0.00	54,360.53	69.21	69.21
Routing-only	0.60	0.75	792.06	0.00	0.00	54,479.13	68.78	68.78
Routing-only	0.60	0.60	804.37	0.00	0.00	54,564.10	67.83	67.83

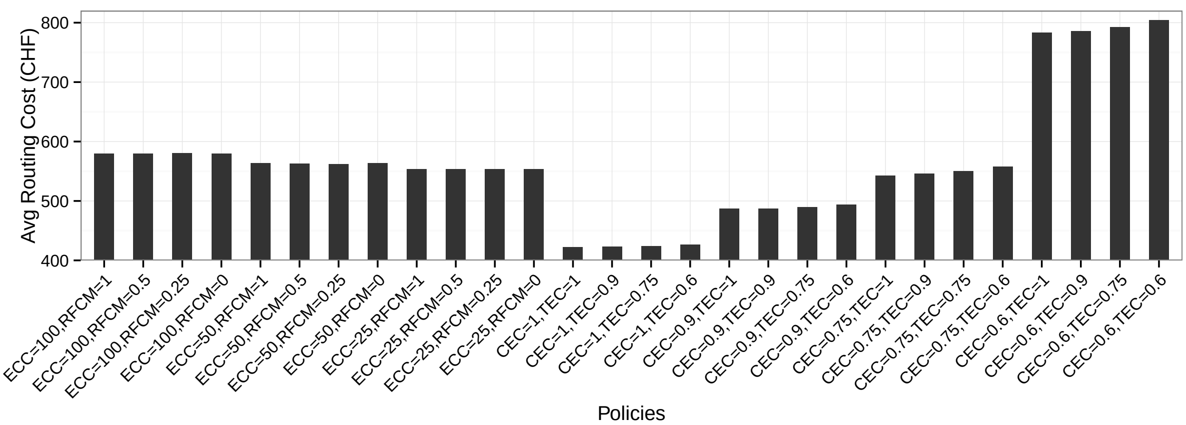
Table 9: Alternative Policies: Container Overflows and Route Failures for Real Data Instances

Objective	CEC	TEC	Avg Num Overflows				Avg Num Route Failures			
			75th Perc.	90th Perc.	95th Perc.	99th Perc.	75th Perc.	90th Perc.	95th Perc.	99th Perc.
Routing-only	1.00	1.00	16.96	20.47	22.58	26.71	0.03	0.05	0.05	0.05
Routing-only	1.00	0.90	16.93	20.42	22.54	26.68	0.00	0.00	0.00	0.00
Routing-only	1.00	0.75	16.90	20.42	22.55	26.70	0.00	0.00	0.00	0.00
Routing-only	1.00	0.60	16.85	20.37	22.50	26.63	0.00	0.00	0.00	0.00
Routing-only	0.90	1.00	10.29	13.07	14.78	18.23	0.02	0.02	0.02	0.02
Routing-only	0.90	0.90	10.25	13.04	14.74	18.15	0.00	0.00	0.00	0.00
Routing-only	0.90	0.75	10.27	13.03	14.77	18.15	0.00	0.00	0.00	0.00
Routing-only	0.90	0.60	10.28	13.02	14.77	18.21	0.00	0.00	0.00	0.00
Routing-only	0.75	1.00	4.23	6.07	7.25	9.65	0.06	0.06	0.06	0.06
Routing-only	0.75	0.90	4.25	6.06	7.27	9.66	0.00	0.00	0.00	0.00
Routing-only	0.75	0.75	4.25	6.07	7.29	9.68	0.00	0.00	0.00	0.00
Routing-only	0.75	0.60	4.24	6.03	7.25	9.67	0.00	0.00	0.00	0.00
Routing-only	0.60	1.00	2.17	3.52	4.45	6.34	0.01	0.01	0.01	0.01
Routing-only	0.60	0.90	2.18	3.52	4.48	6.32	0.00	0.00	0.00	0.00
Routing-only	0.60	0.75	2.15	3.54	4.46	6.29	0.00	0.00	0.00	0.00
Routing-only	0.60	0.60	2.17	3.53	4.47	6.31	0.00	0.00	0.00	0.00

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 4 and 5 present a side-by-side comparison of the probabilistic and the alternative policies 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 16 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 4 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 420 to 800 CHF, with pronounced jumps linked to the variation of the container effective capacity. We observe a disproportionate cost increase linked to lowering the container effective capacity from 75% to 60%. This effect is due to the fact that many more containers

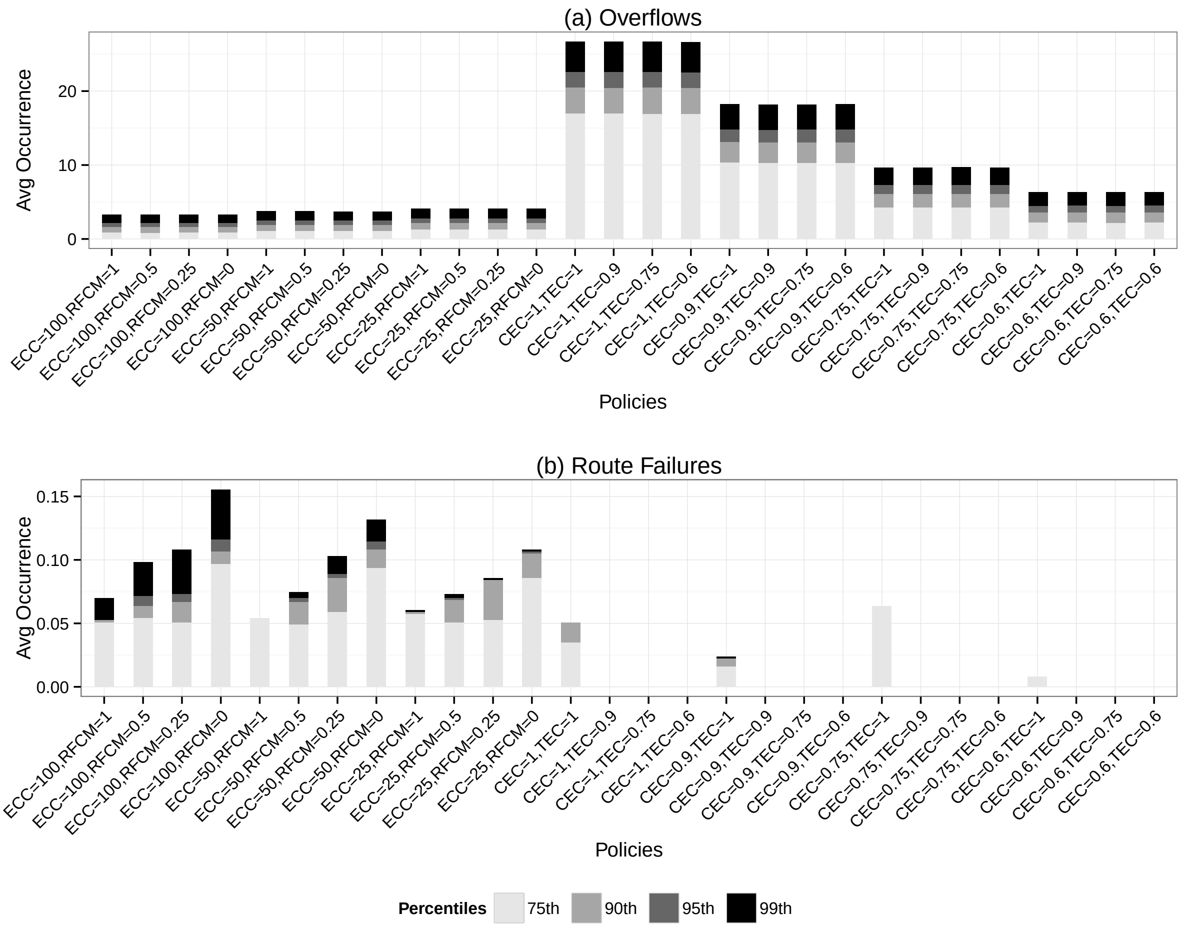
Figure 4: Comparison of Routing Cost for Probabilistic and Alternative Policies



need to be collected now. There are on average three vs. two tours per solution, compared to the case of a CEC of 75% or 90%. Moreover, tours are on average also longer and as a result less compact.

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 5, in parts (a) and (b), respectively. Part (a) of the figure reveals that all considered probabilistic policies are able to limit 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. Nevertheless, the average number of overflows across all instances is approximately four, 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 alternative policies cannot beat the probabilistic ones. A case in point are the complete objective solutions for an ECC of 25 CHF and the routing-only solutions for a CEC of 75%. While they incur a similar routing cost as shown in Figure 4, Figure 5 reveals that the occurrence of overflows for the routing-only solutions is more than twice as high. Reducing further the CEC leads to a mild decrease in the occurrence of overflows accompanied by a significant increase in the routing cost. We stress here that since we compare the performance of two policy types in terms of

Figure 5: Comparison of Container Overflows and Route Failures for Probabilistic and Alternative Policies



number of overflows and route failures at different percentiles, we must isolate these components from the solution cost of the probabilistic model, with a fair cost comparison thus given by the routing cost. These results clearly support the findings of Markov et al. (2018) of the superior performance of the probabilistic policies in the face of stochastic demand.

Lastly, part (b) of Figure 5 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 container visits. Trucks thus collect more containers in each tour and, by consequence, in each depot-to-dump or dump-to-dump trip. 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 5, the contribution of the expected route failure cost to the total cost is immaterial.

5.3.3. Rolling Horizon Experiments.

In practice, the SIRP that we consider is solved in a daily rolling horizon fashion using the newly revealed 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, 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 $\mathcal{T} = \{0, \ldots, u\}$. 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. On average, there are 69 containers per instance, and the maximum is 86. Otherwise, the instances fit the same description as the set of 63 one-week-long instances described in Section 5.2. 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 10 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 for each instance. The 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 61% 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 ς. In other words, if more accurate forecasting methodologies are available, this gap may be brought down substantially. On the other hand, the static SIRP approach is on average 14% more expensive than the rolling horizon approach, clearly showing the benefit of the latter in practical applications.

Table 10: 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	585.69	658.39	Inst_22	429.20	526.06	607.22
Inst_2	**448.67**	**937.47**	**849.43**	Inst_23	241.44	568.15	681.54
Inst_3	307.88	626.01	816.05	**Inst_24**	**547.92**	**769.08**	**747.64**
Inst_4	266.15	577.82	701.61	Inst_25	446.31	583.87	689.37
Inst_5	450.14	663.50	790.44	Inst_26	442.02	575.57	656.27
Inst_6	300.73	624.62	708.79	Inst_27	441.36	595.47	705.01
Inst_7	268.65	580.83	649.67	Inst_28	465.74	628.59	733.80
Inst_8	427.17	608.31	680.36	Inst_29	436.25	579.74	701.33
Inst_9	442.34	609.44	656.44	**Inst_30**	**414.41**	**701.87**	**692.33**
Inst_10	448.70	578.34	647.05	Inst_31	442.87	530.14	668.17
Inst_11	467.35	614.28	669.33	Inst_32	255.32	617.04	695.62
Inst_12	**449.20**	**681.10**	**625.59**	Inst_33	460.04	641.00	773.33
Inst_13	254.66	558.57	629.36	Inst_34	505.55	674.98	710.84
Inst_14	276.60	613.72	685.64	Inst_35	481.85	746.10	786.94
Inst_15	429.26	562.12	788.75	Inst_36	454.60	658.51	741.02
Inst_16	529.60	626.97	702.61	Inst_37	465.33	651.41	749.50
Inst_17	423.07	589.66	663.90	Inst_38	519.56	709.76	809.91
Inst_18	457.65	596.14	681.29	Inst_39	243.94	623.29	697.93
Inst_19	448.66	524.41	596.81	Inst_40	450.94	620.09	756.48
Inst_20	418.12	569.73	653.22	Inst_41	403.01	576.45	688.68
Inst_21	276.32	570.41	622.47				

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

6. Conclusion

We motivate and formulate a real-world stochastic inventory routing problem which includes a range of practical and policy-related constraints. To solve the problem, we develop an ALNS algorithm and use a realistic demand forecasting model. We analyze our stochastic modeling approach on instances derived from real data and demonstrate the relevance of the rich probability information that we model in the objective function. We observe that capturing this information leads to only a moderate increase in the routing cost, while avoiding major expenditures even at relatively lower percentiles of the simulated demand scenarios. Based on our policy, we can control the rate of occurrence of undesirable events, like container overflows and route failures, by scaling their probability-related costs. Our approach significantly outperforms alternative deterministic 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. Finally, we show the benefit of the rolling horizon approach that includes the newly revealed container information each day and derive empirical lower and upper bounds on the its cost.

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. Developing an exact method for solving small instances or a lower bounding procedure for realistic instances can be used to provide benchmark results for the ALNS. Using different modeling techniques, such as scenario generation or robust optimization, can also be used to evaluate the relative quality of the proposed framework. 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 [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. The authors would also like to thank Raphaël Lüthi and Prisca Aeby, both Master's students at the Transport and Mobility Laboratory of École Polytechnique Fédérale de Lausanne, for their dedicated and high-quality work on this problem, and Matthieu de Lapparent, a scientific collaborator at the same laboratory, for his advice on probability and statistics related questions. Declarations of interest: none.

References

Abdollahi, M., Arvan, M., Omidvar, A., & 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*, 603–620. doi:10.5267/j.ijiec.2014.6.003.

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

Adulyasak, Y., Cordeau, J.-F., & Jans, R. (2015). Benders decomposition for production routing under demand uncertainty. *Operations Research*, *63*, 851–867. doi:10.1287/opre.2015.1401.

Angelelli, E., & Speranza, M. G. (2002). The periodic vehicle routing problem with intermediate facilities. *European Journal of Operational Research*, *137*, 233–247. doi:10.1016/S0377-2217(01)00206-5.

Archetti, C., Bertazzi, L., Laporte, G., & Speranza, M. G. (2007). A branch-and-cut algorithm for a vendor-managed inventory-routing problem. *Transportation Science*, *41*, 382–391. doi:10.1287/trsc.1060.0188.

Baldacci, R., & Mingozzi, A. (2009). A unified exact method for solving different classes of vehicle routing problems. *Mathematical Programming*, *120*, 347–380. doi:10.1007/s10107-008-0218-9.

Bard, J. F., Huang, L., Jaillet, P., & Dror, M. (1998). A decomposition approach to the inventory routing problem with satellite facilities. *Transportation Science*, *32*, 189–203. doi:10.1287/trsc.32.2.189.

Beltrami, E. J., & Bodin, L. D. (1974). Networks and vehicle routing for municipal waste collection. *Networks*, *4*, 65–94. doi:10.1002/net.3230040106.

Bertazzi, L., Bosco, A., Guerriero, F., & Laganà, D. (2013). A stochastic inventory routing problem with stock-out. *Transportation Research Part C: Emerging Technologies*, *27*, 89–107. doi:10.1016/j.trc.2011.06.003.

Bertazzi, L., Bosco, A., & Laganà, D. (2015). Managing stochastic demand in an inventory routing problem with transportation procurement. *Omega*, *56*, 112–121. doi:10.1016/j.omega.2014.09.010.

Bertsimas, D., & Sim, M. (2003). Robust discrete optimization and network flows. *Mathematical Programming*, *98*, 49–71. doi:10.1007/s10107-003-0396-4.

Bertsimas, D., & Sim, M. (2004). The price of robustness. *Operations Research*, *52*, 35–53. doi:10.1287/opre.1030.0065.

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., & 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. doi:10.1016/j.sbspro.2012.03.105.

Campbell, A. M., & Savelsbergh, M. W. P. (2004). A decomposition approach for the inventory-routing problem. *Transportation Science*, *38*, 488–502. doi:10.1287/trsc.1030.0054.

Coelho, L. C., Cordeau, J.-F., & Laporte, G. (2012). Consistency in multi-vehicle inventory-routing. *Transportation Research Part C: Emerging Technologies*, *24*, 270–287. doi:10.1016/j.trc.2012.03.007.

Coelho, L. C., Cordeau, J.-F., & Laporte, G. (2014a). Heuristics for dynamic and stochastic inventory-routing. *Computers & Operations Research*, *52*, 55–67. doi:10.1016/j.cor.2014.07.001.

Coelho, L. C., Cordeau, J.-F., & Laporte, G. (2014b). Thirty years of inventory routing. *Transportation Science*, *48*, 1–19. doi:10.1287/trsc.2013.0472.

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

Crevier, B., Cordeau, J.-F., & Laporte, G. (2007). The multi-depot vehicle routing problem with inter-depot routes. *European Journal of Operational Research*, *176*, 756–773. doi:10.1016/j.ejor.2005.08.015.

Dror, M., & Ball, M. (1987). Inventory/routing: Reduction from an annual to a short-period problem. *Naval Research Logistics*, *34*, 891–905. doi:10.1002/1520-6750(198712)34:6<891::AID-NAV3220340613>3.0.CO;2-J.

Dror, M., & Trudeau, P. (1996). Cash flow optimization in delivery scheduling. *European Journal of Operational Research*, *88*, 504–515. doi:10.1016/0377-2217(94)00205-3.

Erdoğan, S., & Miller-Hooks, E. (2012). A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, *48*, 100–114. doi:10.1016/j.tre.2011.08.001.

European Commission (2016). *Circular Economy Strategy.* URL: http://ec.europa.eu/environment/circular-economy Accessed on April 16, 2018.

Hemmelmayr, V., Doerner, K. F., Hartl, R. F., & Rath, S. (2013). A heuristic solution method for node routing based solid waste collection problems. *Journal of Heuristics*, *19*, 129–156. doi:10.1007/s10732-011-9188-9.

Hemmelmayr, V., Doerner, K. F., Hartl, R. F., & Savelsbergh, M. W. (2010). Vendor managed inventory for environments with stochastic product usage. *European Journal of Operational Research*, *202*, 686–695. doi:10.1016/j.ejor.2009.06.003.

Jaillet, P., Bard, J. F., Huang, L., & Dror, M. (2002). Delivery cost approximations for inventory routing problems in a rolling horizon framework. *Transportation Science*, *36*, 292–300. doi:10.1287/trsc.36.3.292.7829.

Kim, B. I., Kim, S., & Sahoo, S. (2006). Waste collection vehicle routing problem with time windows. *Computers & Operations Research*, *33*, 3624–3642. doi:10.1016/j.coe.2005.02.045.

Markov, I., Bierlaire, M., Cordeau, J.-F., Maknoon, Y., & Varone, S. (2018). A unified framework for rich routing problems with stochastic demands. *Transportation Research Part B: Methodological*, *114*, 213–240. doi:10.1016/j.trb.2018.05.015.

Markov, I., de Lapparent, M., Bierlaire, M., & 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., & 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. doi:10.1016/j.trb.2015.12.004.

Mes, M., Schutten, M., & Rivera, A. P. (2014). Inventory routing for dynamic waste collection. *Waste Management*, *34*, 1564–1576. doi:10.1016/j.wasman.2014.05.011.

Moin, N. H., & Salhi, S. (2007). Inventory routing problems: A logistical overview. *Journal of the Operational Research Society*, *58*, 1185–1194. doi:10.1057/palgrave.jors.2602264.

Muter, I., Cordeau, J.-F., & Laporte, G. (2014). A branch-and-price algorithm for the multidepot vehicle routing problem with interdepot routes. *Transportation Science*, *48*, 425–441. doi:10.1287/trsc.2013.0489.

Nolz, P. C., Absi, N., & Feillet, D. (2014). A stochastic inventory routing problem for infectious medical waste collection. *Networks*, *63*, 82–95. doi:10.1002/net.21523.

Pelletier, S., Jabali, O., & Laporte, G. (2016). 50th anniversary invited article–Goods distribution with electric vehicles: Review and research perspectives. *Transportation Science*, *50*, 3–22. doi:10.1287/trsc.2015.0646.

Penna, P. H. V., Subramanian, A., & Ochi, L. S. (2013). An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. *Journal of Heuristics*, *19*, 201–232. doi:10.1007/s10732-011-9186-y.

Rahbari, A., Nasiri, M. M., & Ghorbani, E. (2017). An inventory routing problem for perishable products with demand and transportation time uncertainty: A robust optimization approach. In *Proceedings of the 10th International Conference of the Iranian Operations Research Society*. May 3–5, 2017, Babolsar, Iran.

Roldán, R. F., Basagoiti, R., & Coelho, L. C. (2016). Robustness of inventory replenishment and customer selection policies for the dynamic and stochastic inventory-routing problem. *Computers & Operations Research*, *74*, 14–20. doi:10.1016/j.cor.2016.04.004.

Ropke, S., & Pisinger, D. (2006a). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, *40*, 455–472. doi:10.1287/trsc.1050.0135.

Ropke, S., & Pisinger, D. (2006b). A unified heuristic for a large class of vehicle routing problems with backhauls. *European Journal of Operational Research*, *171*, 750–775. doi:10.1016/j.ejor.2004.09.004.

Savelsbergh, M. W. P. (1992). The vehicle routing problem with time windows: Minimizing route duration. *INFORMS Journal on Computing*, *4*, 146–154. doi:10.1287/ijoc.4.2.146.

Schneider, M., Stenger, A., & Goeke, D. (2014). The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science*, *48*, 500–520. doi:10.1287/trsc.2013.0490.

Schneider, M., Stenger, A., & Hof, J. (2015). An adaptive VNS algorithm for vehicle routing problems with intermediate stops. *OR Spectrum*, *37*, 353–387. doi:10.1007/s00291-014-0376-5.

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., & Laporte, G. (2012). Robust inventory routing under demand uncertainty. *Transportation Science*, *46*, 327–340. doi:10.1287/trsc.1110.0387.

Soysal, M., Bloemhof-Ruwaard, J. M., Haijema, R., & 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. doi:10.1016/j.ijpe.2015.03.008.

Soysal, M., Bloemhof-Ruwaard, J. M., Haijema, R., & van der Vorst, J. G. (2018). Modeling a green inventory routing problem for perishable products with horizontal collaboration. *Computers & Operations Research*, *89*, 168–182. doi:10.1016/j.cor.2016.02.003.

Subramanian, A., Penna, P. H. V., Uchoa, E., & Ochi, L. S. (2012). A hybrid algorithm for the heterogeneous fleet vehicle routing problem. *European Journal of Operational Research*, *221*, 285–295. doi:10.1016/j.ejor.2012.03.016.

Taillard, É. D. (1999). A heuristic column generation method for the heterogeneous fleet VRP. *RAIRO - Operations Research*, *33*, 1–14. doi:10.1051/ro:1999101.

Tavares, G., Zsigraiova, Z., Semiao, V., & Carvalho, M. (2009). Optimisation of MSW collection routes for minimum fuel consumption using 3D GIS modelling. *Waste Management*, *29*, 1176–1185. doi:10.1016/j.wasman.2008.07.013.

Trudeau, P., & Dror, M. (1992). Stochastic inventory routing: Route design with stockouts and route failures. *Transportation Science*, *26*, 171–184. doi:10.1287/trsc.26.3.171.

Yu, Y., Chu, C., Chen, H., & Chu, F. (2012). Large scale stochastic inventory routing problems with split delivery and service level constraints. *Annals of Operations Research*, *197*, 135–158. doi:10.1007/s10479-010-0772-4.

Waste Collection Inventory Routing with Non-stationary Stochastic Demands: Online Appendices

Iliya Markov[a,*], Michel Bierlaire[a], Jean-François Cordeau[b], Yousef Maknoon[c], Sacha Varone[d]

[a] *Transport and Mobility Laboratory*
School of Architecture, Civil and Environmental Engineering
École Polytechnique Fédérale de Lausanne
Station 18, 1015 Lausanne, Switzerland

[b] *CIRRELT and HEC Montréal*
3000 chemin de la Côte-Sainte-Catherine
Montréal, Canada H3T 2A7

[c] *Faculty of Technology, Policy, and Management*
Delft University of Technology
Jaffalaan 5, 2628 BX Delft, The Netherlands

[d] *Haute École de Gestion de Genève*
University of Applied Sciences Western Switzerland (HES-SO)
Campus Battelle, Rue de la Tambourine 17, 1227 Carouge, Switzerland

Appendix A. Benchmark Formulations

Appendix A.1. IRP Benchmark Formulation

The IRP benchmark instances 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 $\rho_{it} = \mathbb{E}(\rho_{it})$. A 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 resulting problem is composed of the inventory holding costs at the depot and the customers, and the routing cost, and writes as:

$$\min z^{\text{IRPB}} = \sum_{t \in \mathcal{T} \cup \mathcal{T}^+} h_o I_{ot} + \sum_{t \in \mathcal{T} \cup \mathcal{T}^+} \sum_{i \in \mathcal{P}} h_i I_{it} + \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} \left(\varphi_k z_{kt} + \beta_k \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} \pi_{ij} x_{ijkt} + \theta_k \left(S_{dkt} - S_{okt} \right) \right). \quad \text{(A.1)}$$

A special set of constraints is needed for the inventory definition at the depot. Constraints (A.2) 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. Constraints (A.3) forbid a stock-out at the depot given the total quantity delivered to customers.

$$I_{ot} = I_{o(t-1)} + \rho_{o(t-1)} - \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{P}} q_{ik(t-1)}, \qquad \forall t \in \mathcal{T}^+ \quad \text{(A.2)}$$

*Corresponding author

Email addresses: iliya.markov@epfl.ch (Iliya Markov), michel.bierlaire@epfl.ch (Michel Bierlaire), jean-francois.cordeau@hec.ca (Jean-François Cordeau), m.y.maknoon@tudelft.nl (Yousef Maknoon), sacha.varone@hesge.ch (Sacha Varone)

$$I_{ot} \geqslant \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{P}} q_{ikt}, \qquad \forall t \in \mathcal{T} \tag{A.3}$$

We redefine the evolution of the inventory level at the customers for a distribution context. Constraints (A.4) 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 (A.5) forbid the occurrence of stock-outs.

$$I_{it} = I_{i(t-1)} + \sum_{k\in\mathcal{K}} q_{ik(t-1)} - \rho_{i(t-1)}, \qquad \forall i \in \mathcal{P}, t \in \mathcal{T}^+ \tag{A.4}$$

$$I_{it} \geqslant 0, \qquad \forall i \in \mathcal{P}, t \in \mathcal{T}^+ \tag{A.5}$$

The order-up-to policy also needs to be redefined for a distribution context. Constraints (A.6), (A.7), and (A.8) express the fact that if a customer is visited, its inventory is brought up to its capacity.

$$q_{ikt} \geqslant \omega_i y_{ikt} - I_{it}, \qquad \forall t \in \mathcal{T}, k \in \mathcal{K}, i \in \mathcal{P} \tag{A.6}$$

$$q_{ikt} \leqslant \omega_i - I_{it}, \qquad \forall t \in \mathcal{T}, k \in \mathcal{K}, i \in \mathcal{P} \tag{A.7}$$

$$q_{ikt} \leqslant \omega_i y_{ikt}, \qquad \forall t \in \mathcal{T}, k \in \mathcal{K}, i \in \mathcal{P} \tag{A.8}$$

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)–(30), the intra-day temporal constraints (31)–(35), and the domain constraints (36)–(37) can thus be reused.

Appendix A.2. VRP Benchmark Formulation

For the VRP benchmark instances, 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:

$$\min z^{\text{VRPB}} = \sum_{k\in\mathcal{K}} \left(\varphi_k z_{k0} + \beta_k \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} \pi_{ij} x_{ijk0} + \theta_k \left(S_{dk0} - S_{ok0} \right) \right). \tag{A.9}$$

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, establishing that each container should be 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.

Appendix B. ALNS Modifications

Appendix B.1. Modifications for Solving IRP Benchmarks

The IRP benchmark instances consider 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 (A.3) in the IRP benchmark formulation:

$$V^I(s) = \sum_{t\in\mathcal{T}} \left(\sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{P}} q_{ikt} - I_{ot} \right)^+. \tag{B.1}$$

The violation $V^I(s)$ is multiplied by parameter L^I and added to the objective function representation $f(s)$ as in expression (45). Additionally, we redefine the container capacity violation (42) in terms of stock-out as it applies to a distribution context:

$$V^\omega(s) = \sum_{t\in\mathcal{T}^+} \sum_{i\in\mathcal{P}} (-I_{it})^+. \tag{B.2}$$

The back-order violation (43) is dropped since back-orders do not apply to the IRP benchmark formulation.

Appendix B.2. Modifications for Solving VRP Benchmarks

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 5 through 10 are now drawn first, while repair operators 1 through 4 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*, *insert ν containers with regret-k*, and *Shaw insertions with relatedness*. 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 (42) and (43) 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.

Appendix C. Benchmark Results

Sections Appendix C.1 and Appendix C.2 below present the results obtained by the ALNS on classical IRP and VRP instance testbeds. Archetti et al.'s (2007) instances test the ALNS in a classical IRP context with order-up-to level policy. Crevier et al.'s (2007) instances test the ALNS in a VRP setting with intermediate facilities, the latter being an important feature that is present in our problem in the form of dumps. Crevier et al.'s (2007) instances are also representative of a standard waste collection problem and used by Hemmelmayr et al. (2013) in this context. Finally, Taillard's (1999) instances test the ALNS in a VRP context with a rich heterogeneous fixed fleet, which is the most general type of fleet. These benchmark instances capture many of the features of our waste collection IRP, albeit not simultaneously.

Appendix C.1. Results on IRP Benchmarks

The IRP benchmark set proposed by Archetti et al. (2007) is the first classical IRP testbed in the literature and represents a deterministic IRP in a distribution context. 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 order-up-to 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.

Table C.1 presents our results on the Archetti et al. (2007) instances. The first two columns report the number of periods $|\mathcal{T}|$ in the planning horizon and the number of customers n. The remainder of the table is divided into two parts, with results for the instances with high, respectively low, inventory holding cost, each providing the average runtime in seconds, and the best, average and worst gap to the optimal solution obtained over 10 runs. Each row averages the latter over the five instances for each combination of h_i, $|\mathcal{T}|$ and n. Our results are comparable to the best from the literature. The ALNS attains a best gap of 0.02% and 0.05%, an average gap of 0.15% and 0.39%, and a worst gap of 0.34% and 0.89% on the high and low inventory holding cost instances, respectively. In comparison, Archetti et al.'s (2012) algorithm obtains a gap of 0.06% and 0.10%, respectively, for a single run per instance. We are able to solve to optimality almost all instances with up to 35 customers for $|\mathcal{T}| = 3$, and with

Table C.1: Results on Archetti et al. (2007) Instances

		High Inventory Holding Cost				Low Inventory Holding Cost			
$\|\mathcal{T}\|$	n	Runtime(s.)	Best Gap(%)	Avg Gap(%)	Worst Gap(%)	Runtime(s.)	Best Gap(%)	Avg Gap(%)	Worst Gap(%)
3	5	69.08	0.00	0.00	0.00	85.69	0.00	0.00	0.00
3	10	183.94	0.00	0.00	0.00	156.36	0.00	0.00	0.00
3	15	317.93	0.00	0.00	0.00	274.05	0.00	0.00	0.00
3	20	440.02	0.00	0.00	0.01	444.68	0.00	0.00	0.02
3	25	523.42	0.00	0.08	0.25	501.78	0.01	0.20	0.66
3	30	835.21	0.01	0.15	0.32	649.09	0.00	0.41	0.98
3	35	866.06	0.00	0.15	0.36	731.21	0.00	0.46	1.68
3	40	896.91	0.02	0.18	0.44	976.83	0.16	0.47	0.97
3	45	1124.57	0.05	0.42	0.91	1074.19	0.00	1.05	2.53
3	50	1424.27	0.06	0.35	0.79	1223.56	0.13	1.19	2.15
6	5	105.86	0.00	0.00	0.00	73.28	0.00	0.00	0.00
6	10	184.48	0.00	0.01	0.08	181.93	0.00	0.00	0.00
6	15	333.82	0.01	0.09	0.15	272.03	0.00	0.03	0.16
6	20	394.39	0.00	0.17	0.41	420.28	0.05	0.34	0.82
6	25	636.27	0.12	0.34	0.82	546.85	0.09	0.67	1.60
6	30	725.63	0.10	0.47	0.93	733.12	0.44	1.43	2.63
Average		566.37	0.02	0.15	0.34	521.56	0.05	0.39	0.89

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.

Appendix C.2. Results on VRP Benchmarks

The SIRP that we study includes a rich routing component. Since the routing component in the Archetti et al. (2007) IRP instances is very simple, we perform further tests two VRP benchmark 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 C.2, the instance name is followed by the number of customers n, the computation time in seconds, the best and average cost obtained by Hemmelmayr et al. (2013) over 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 0.40% from those of Hemmelmayr et al. (2013) and we are able to reach five of the BKS. Our gap with respect to the average value over 10 runs is under 1%. We observe that our ALNS is roughly six times slower than Hemmelmayr et al.'s (2013) VNS on these instances.

Table C.2: Results on Crevier et al. (2007) Instances

		Hemmelmayr et al. (2013)			ALNS				
Instance	n	Runtime(s.)	Best Cost	Avg Cost	Runtime(s.)	Best Cost	Avg Cost	Gap Best(%)	Gap Avg(%)
a2	48	73.80	997.94	997.94	140.41	997.94	998.17	0.00	0.02
b2	96	384.60	1291.19	1291.19	681.63	1291.19	1293.11	0.00	0.15
c2	144	900.60	1715.60	1715.84	2314.81	1733.95	1740.11	1.07	1.41
d2	192	1808.40	1856.84	1860.92	9264.81	1870.99	1879.90	0.76	1.02
e2	240	2958.60	1919.38	1922.81	17,694.38	1929.05	1943.05	0.50	1.05
f2	288	4274.40	2230.32	2233.43	32,170.43	2247.60	2273.39	0.77	1.79
g2	72	222.60	1152.92	1153.17	475.02	1152.92	1153.15	0.00	0.00
h2	144	939.60	1575.28	1575.28	2496.57	1582.78	1587.06	0.48	0.75
i2	216	2515.20	1919.74	1922.24	13,896.84	1947.78	1964.21	1.46	2.18
j2	288	4402.80	2247.70	2250.21	40,936.64	2259.25	2283.23	0.51	1.47
a1	48	85.20	1179.79	1180.57	200.24	1190.12	1197.22	0.88	1.41
b1	96	383.40	1217.07	1217.07	925.46	1218.09	1218.97	0.08	0.16
c1	192	1224.00	1866.76	1867.96	6435.66	1874.47	1884.54	0.41	0.89
d1	48	94.20	1059.43	1059.43	139.35	1059.43	1061.86	0.00	0.23
e1	96	373.20	1309.12	1309.12	743.72	1309.12	1319.97	0.00	0.83
f1	192	1536.00	1570.41	1573.05	7303.54	1571.62	1587.91	0.08	0.94
g1	72	202.80	1181.13	1183.32	308.81	1185.99	1189.76	0.41	0.54
h1	144	876.60	1545.50	1548.61	2622.40	1553.12	1565.19	0.49	1.07
i1	216	2014.80	1922.18	1923.52	10,383.43	1925.19	1935.39	0.16	0.62
j1	72	166.80	1115.78	1115.78	489.83	1116.67	1123.14	0.08	0.66
k1	144	873.60	1576.36	1577.96	2937.01	1580.84	1591.56	0.28	0.86
l1	216	2128.80	1863.28	1869.70	9343.72	1870.87	1884.84	0.41	0.81
Average		1292.73	1559.71	1561.32	7359.30	1566.77	1576.17	0.40	0.86

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 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 & Mingozzi (2009). In Table C.3, the instance name is followed by the number of customers n and the BKS, which are due to multiple authors. Next are the computation time in seconds, the best, average and worst cost obtained by our ALNS. The last three columns report the percent gap of our best, average and worst cost with respect to the BKS. Our results are on average 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 30 minutes. In sum, the numerical experiments show that our ALNS exhibits very good performance on the considered

Table C.3: Results on Taillard (1999) Instances

			ALNS						
Instance	n	BKS	Runtime(s.)	Best Cost	Avg Cost	Worst Cost	Best Gap(%)	Avg Gap(%)	Worst Gap(%)
13	50	3185.09	317.91	3228.97	3250.87	3279.00	1.38	2.07	2.95
14	50	10,107.53	335.92	10,123.50	10,143.21	10,158.54	0.16	0.35	0.50
15	50	3065.29	318.48	3072.59	3082.41	3090.24	0.24	0.56	0.81
16	50	3265.41	326.63	3292.22	3312.42	3322.43	0.82	1.44	1.75
17	75	2076.96	762.52	2097.29	2132.52	2148.15	0.98	2.68	3.43
18	75	3743.58	847.68	3801.22	3832.18	3860.08	1.54	2.37	3.11
19	100	10,420.34	1950.83	10,478.69	10,500.96	10,515.97	0.56	0.77	0.92
20	100	4761.26	1598.25	4901.45	4927.19	4946.61	2.94	3.49	3.89
Average		5078.18	807.27	5124.49	5147.72	5165.13	1.08	1.71	2.17

VRP benchmark instances when compared to state-of-the-art methods developed specifically for these problems.

References

Archetti, C., Bertazzi, L., Hertz, A., & Speranza, M. G. (2012). A hybrid heuristic for an inventory routing problem. *INFORMS Journal on Computing*, *24*, 101–116. doi:10.1287/ijoc.1100.0439.

Archetti, C., Bertazzi, L., Laporte, G., & Speranza, M. G. (2007). A branch-and-cut algorithm for a vendor-managed inventory-routing problem. *Transportation Science*, *41*, 382–391. doi:10.1287/trsc.1060.0188.

Baldacci, R., & Mingozzi, A. (2009). A unified exact method for solving different classes of vehicle routing problems. *Mathematical Programming*, *120*, 347–380. doi:10.1007/s10107-008-0218-9.

Coelho, L. C., Cordeau, J.-F., & Laporte, G. (2012a). Consistency in multi-vehicle inventory-routing. *Transportation Research Part C: Emerging Technologies*, *24*, 270–287. doi:10.1016/j.trc.2012.03.007.

Coelho, L. C., Cordeau, J.-F., & Laporte, G. (2012b). The inventory-routing problem with transshipment. *Computers & Operations Research*, *39*, 2537–2548. doi:10.1016/j.cor.2011.12.020.

Cordeau, J.-F., Gendreau, M., & Laporte, G. (1997). A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, *30*, 105–119. doi:10.1002/(SICI)1097-0037(199709)30:2<105::AID-NET5>3.0.CO;2-G.

Crevier, B., Cordeau, J.-F., & Laporte, G. (2007). The multi-depot vehicle routing problem with inter-depot routes. *European Journal of Operational Research*, *176*, 756–773. doi:10.1016/j.ejor.2005.08.015.

Golden, B., Assad, A., Levy, L., & Gheysens, F. (1984). The fleet size and mix vehicle routing problem. *Computers & Operations Research*, *11*, 49–66. doi:10.1016/0305-0548(84)90007-8.

Hemmelmayr, V., Doerner, K. F., Hartl, R. F., & Rath, S. (2013). A heuristic solution method for node routing based solid waste collection problems. *Journal of Heuristics*, *19*, 129–156. doi:10.1007/s10732-011-9188-9.

Taillard, É. D. (1999). A heuristic column generation method for the heterogeneous fleet VRP. *RAIRO - Operations Research*, *33*, 1–14. doi:10.1051/ro:1999101.