CIRRELT

Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

A Rule-Based Recourse for the Vehicle Routing Problem with Stochastic Demands

Majid Salavati-Khoshghalb
Michel Gendreau
Ola Jabali
Walter Rei

June 2017

CIRRELT-2017-36

Bureaux de Montréal :
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

UNIVERSITÉ LAVAL McGill Université Concordia University ÉTS UQÀM Université du Québec à Montréal HEC MONTRÉAL POLYTECHNIQUE MONTRÉAL Université de Montréal

A Rule-Based Recourse for the Vehicle Routing Problem with Stochastic Demands

Majid Salavati-Khoshghalb[1,2,*], Michel Gendreau[1,3], Ola Jabali[4], Walter Rei[1,5]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[2] Department of Computer Science and Operations Research, Université de Montréal, P.O. Box 6128, Station Centre-Ville, Montréal, Canada H3C 3J7

[3] Department of Mathematics and Industrial Engineering, Polytechnique Montréal, P.O. Box 6079, Station Centre-Ville, Montréal, Canada H3C 3A7

[4] Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Piazza Leonardo da Vinci, 32, Milano 20133, Italy

[5] Department of Management and Technology, Université du Québec à Montréal, P.O. Box 8888, Station Centre-Ville, Montréal, Canada H3C 3P8

Abstract. In this paper we consider the vehicle routing problem with stochastic demands (VRPSD). We consider that customer demands are only revealed when a vehicle arrives at customer locations. Failures occur whenever the residual capacity of the vehicle is insufficient to serve the observed demand of a customer. Such failures entail that recourse actions be taken to recover route feasibility. These recourse actions usually take the form of return trips to the depot, which can be either done in a reactive or proactive fashion. Over the years, there have been various policies defined to perform these recourse actions in either a static or a dynamic setting. In the present paper, we propose policies that better reflect the fixed operational rules that can be observed in practice, and that also enable implementing preventive recourse actions. We define the considered operational rules and show how, for a planned route, these operational rules can be implemented using a fixed threshold-based policy to govern the recourse actions. An exact solution algorithm is developed to solve the VRPSD under the considered policies. Finally, we conduct an extensive computational study, which shows that significantly better solutions can be obtained when using the proposed policies compared to solving the problem under the classical recourse definition.

Keywords: Threshold-based recourse policies, operational rules, vehicle routing problem with stochastic demands, partial routes, Integer L-shaped algorithm, lower bounding functionals.

Acknowledgements. This research was partially funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) under the discovery grants program. This support is gratefully acknowledged.

Results and views expressed in this publication are the sole responsibility of the authors and do not necessarily reflect those of CIRRELT.

Les résultats et opinions contenus dans cette publication ne reflètent pas nécessairement la position du CIRRELT et n'engagent pas sa responsabilité.

---

* Corresponding author: majid.salavati @cirrelt.net

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2017

© Salavati-Khoshghalb, Gendreau, Jabali, Rei and CIRRELT, 2017

1 Introduction

Since the seminal paper of Dantzig and Ramser (1959), thousands of papers have been published on the vehicle routing problem (VRP), which is central to distribution activities. In its simplest version, the VRP consists in designing a set of routes, starting and ending at a given depot location, to serve a set of customers with known demands by a fleet of identical vehicles of finite capacity, with the objective of minimizing the total distance traveled. In the deterministic version of the problem, which has been widely studied, all problem parameters are known precisely and each customer must be visited exactly once (see Toth and Vigo (2014) for a thorough overview of the problem and its main variants). In reality, however, routing problems involve several sources of uncertainty: demands, travel and service times, etc. Routing problems in which some parameters are uncertain are called *Stochastic VRPs* (SVRPs). Although, deterministic approximation models can be solved as proxies for SVRP models, such approximations generally lead to arbitrarily bad solutions, see Louveaux (1998). Therefore, there is a need to develop specialized optimization models that explicitly account for the stochastic nature of VRPs. While they have received much less attention than deterministic VRPs, SVRPs have nonetheless been investigated by several authors; see Gendreau et al. (2014) for a survey of the SVRP literature.

In this paper, we focus on a variant of the SVRP in which customer demands are uncertain. In this variant, which is called the *vehicle routing problem with stochastic demands* (VRPSD), the demand of each customer is assumed to follow a known, customer-specific probability distribution. It is further assumed that each customer's demand is revealed upon the arrival of a vehicle at its location. When demands are stochastic, one could obviously plan routes in such a way that they can handle the maximum possible demand of each customer assigned to it, but in almost all cases, this is extremely inefficient and often times infeasible in terms of the available number of vehicles. To circumvent this difficulty, optimization approaches relying on different modeling paradigms have been proposed (see Gendreau et al. (2014) for a thorough discussion of these paradigms). In this paper, we adopt the *a priori optimization* paradigm, originally proposed by Bertsimas et al. (1990). In this approach, the problem is decomposed into two stages, as in two-stage stochastic programming with recourse. In the first-stage, an *a priori* solution (i.e., a complete set of routes as in a deterministic VRP) is planned. Then, in the second-stage, this first-stage solution is "executed", i.e., each route is followed and the actual values of the uncertain parameters (the customer demands in the case of VRPSD) are gradually revealed.

In the second-stage of the problem, failures may be observed when a route is executed. Such failures occur when the vehicle performing the route arrives at a customer's location without sufficient residual capacity to service the observed demand. These occurrences are simply referred to as *route failures*, see Dror and Trudeau (1986). To recover route feasibility, recourse actions must be taken. As presented in Gendreau et al. (2014), various studies have been conducted to formulate and assess the efficiency of the possible recourse actions that can be applied to the VRPSD.

In the present paper, we focus on the recourse actions that can be implemented independently by the vehicles performing the routes determined in the first-stage of the problem. These recourse actions can either be reactive (i.e., implemented only after a route failure occurs) or proactive (i.e., made in anticipation of possible failures that could take place along the route). A reactive recourse action takes the form of a *back-and-forth* (BF) trip to the depot, where the vehicle is able to restock and then serve the remaining demand at the customer location where the failure occurred. Following a BF trip, the vehicle simply proceeds to the next scheduled customer on the route. In the case of an *exact stockout*, where the revealed demand matches exactly the residual capacity of the vehicle, a *restocking trip* is performed, entailing that the vehicle visits the depot before proceeding to the next customer along the route, see Gendreau et al. (1995) and Hjorring and Holt (1999). In an effort to simplify the presentation of the concepts proposed in this paper, we will refer to BF trips as all reactive recourse actions taken following route failures, be it as the consequence of insufficient residual capacity or an exact stockout. Finally, to avoid route failures, a vehicle may execute a *preventive restocking* (PR) trip whenever its residual capacity becomes too low, see Yee and Golden (1980) and Yang et al. (2000). Considering that such recourse actions are applied before an actual failure is observed, they are regarded as being proactive.

To formulate the VRPSD, a policy, which governs how the recourse actions are applied, must be determined. While a wide variety of recourse policies can be envisioned (see Gendreau et al. (2016)), research has been performed primarily on two categories of recourse actions. In the case where only reactive recourse actions are considered, the *classical recourse* policy is used to model the VRPSD. Following this policy each route is executed until it either fails or faces an exact stockout, at which point an appropriate reactive recourse action is implemented. Several authors have considered this policy and proposed exact solution procedures (e.g., Laporte et al. (2002), Christiansen and Lysgaard (2007), Gauvin et al. (2014), and Jabali et al. (2014)) and heuristics (e.g., Gendreau et al. (1996), Rei et al. (2010), and Mendoza and Villegas (2013)) to solve the resulting model. As an alternative to the classical recourse policy for the VRPSD, Yang et al. (2000) showed that an *optimal restocking policy* can be derived for a given route using dynamic programming. Such a policy takes the form of customer-specific thresholds that, when compared to the residual capacity of the vehicle leaving the customers along the route, specify when a PR trip should be performed. Thus, in Yang et al. (2000), given a route, these customer-specific thresholds are optimized to yield the minimum route cost. It should be noted that, in this case, BF trips are still implemented when failures occur. However, by applying PR trips, the risk of observing route failures is reduced. This approach to formulate the VRPSD coupled with suitable heuristics or metaheuristics to design the a priori routes, was shown to yield more cost-effective solutions, see Bertsimas et al. (1995), Yang et al. (2000) and Bianchi (2006).

The use of both the classical recourse or the optimal restocking policies implies that, in the first-stage of the model, the routing decisions be made statically (i.e., a set of a priori fixed routes are obtained). However, both the routing and recourse decisions (i.e., BF and PR trips) can also be made dynamically. In this case, the VRPSD is formulated using the reoptimization approach, see Secomandi (2001), Novoa and Storer (2009) and Secomandi and Margot (2009). It should be noted that, if reoptimization is applied, the VRPSD is no longer formulated as a two-stage stochastic model. Instead, it can be expressed as a Markov Decision Process, see Dror et al. (1989), or it can be modelled as a stochastic shortest path problem, as detailed in Secomandi (2000).

As a formulation paradigm applied to the VRPSD, the a priori approach is applicable in cases where an organization facing the problem aims to achieve a high level of consistency in its routing operations. Hence, a set of fixed a priori routes are determined, which can then be easily repeated on a daily basis. While the classical recourse policy meets these criteria, its implementation is likely to be costly. The optimal restocking policy provides a better theoretical alternative, however its solution is challenging. Existing heuristics for this policy may exactly evaluate the recourse cost of a given route, however the overall quality of the solutions is not guaranteed. Moreover, many companies employ preset operational conventions when operating in uncertain environments. These are translated into preset rules, which streamline the operations in a manner that greatly simplifies recourse policies. Preset rules can be implemented as a set of fixed rule-based policies. Therefore, we propose a fixed rule-based policy for the VRPSD, according to which the PR trips are governed by preset rules which establish customer-specific thresholds. A detailed motivation for the use of rule-based policies in the VRPSD is provided in Section 2.

In the present paper, we introduce the concept of a rule-based recourse policy for the VRPSD and provide its formulation. We propose an exact solution algorithm for a particular family of volume rule-based recourse policies. We note that to-date exact algorithms for the VRPSD have only been proposed for the VRPSD with classical recourse (e.g., see Gauvin et al. (2014) and Jabali et al. (2014) for recent studies). Finally, by performing an extensive computational study, we demonstrate that significantly better solutions can be obtained using the proposed policies when compared to the classical recourse one, while remaining cost-effective with regards to optimal restocking.

The remainder of this paper is organized as follows. Section §2 discusses general motivations for using rule-based policies in the context of VRPSD. Section §3 lays out the model using a rule-based recourse, then three volume-based rules are defined. Section §4 is devoted to presenting an exact solution methodology to solve the VRPSD under these rules. Various lower bounding procedures are developed to enhance the efficiency of proposed algorithm. Section §5 is dedicated to numerical results and compares rule-based policies in different aspects. Section §6 summarizes the contribution of the paper and points out some future work.

2 Motivation for Rule-Based Policies

In this section, we present the general ideas and observations that warranted the present work. As we will detail, the proposed rule-based recourse approach for the VRPSD is motivated by both practical and methodological considerations. In recent years, the concept of consistency in VRPs has been proposed to improve the overall quality of the service that companies provide to their customers. As presented in Kovacs et al. (2014), there are three dimensions to consistency in the VRP context: 1) arrival time consistency (i.e., customers are visited at approximately the same time whenever deliveries, or pickups, are performed); 2) person-oriented consistency (i.e., customers are assigned to specific drivers that perform the services whenever they are required); 3) delivery consistency (i.e., the actual quantities that are delivered, or collected, reflect the demands of the customers). In the VRPSD literature delivery consistency is predominantly ensured. However, depending on which modelling paradigm is adopted, the first two consistency dimensions may not be guaranteed. In the previously discussed reoptimization paradigm both the routing and the recourse decisions are made dynamically. Therefore, time consistency is not guaranteed. Moreover, person-oriented consistency, may not be enforced if the customers are not clustered and assigned to drivers beforehand.

The a priori paradigm for the VRPSD is a suitable strategy for practical settings where consistency is an important factor. This paradigm guarantees delivery consistency. Moreover, the assumption that vehicles independently perform routes entails that person-oriented consistency is preserved. By allowing PR trips to be performed as part of the recourse decisions, one can further reduce the risk of observing costly failures that significantly lengthen the actual routes that are performed, thus causing arrival time consistency issues.

Using optimal restocking policies for the VRPSD entails using customer-specific thresholds, which are optimized as function of a route. This leaves little control for companies to systematically adjust the customer-specific thresholds. As such, optimal restocking may not reflect a company's operational policies and does not allow it to control the risk of encountering failures. To govern when PR trips are applied, companies may consider a specific set of controllable preset rules to perform the PR trips, e.g., executing a PR trip once the available vehicle capacity is below a preset percentage of its total capacity. Such fixed rules are defined to reflect the overall operational conventions of a company, they preserve consistency and simplify the implementation of the routing plan. As we will detail in the present paper, the ruled-based recourse approach that is developed offers an efficient way to both formulate and apply such fixed rules in the context of the VRPSD.

There are also methodological considerations that motivate the use of the proposed ruled-based recourse. Under the classical recourse policy, the problem of finding a set of a priori routes for the VRPSD is already a complex combinatorial problem (i.e., NP-hard). When PR trips are introduced in the definition of the recourse, this complexity is only compounded. As reported in Yang et al. (2000), solving the dynamic program to obtain an optimal restocking policy for a given route, becomes numerically intractable for routes involving more than 15 customers, which considerably limits the applicability of this approach to practical settings. Therefore, as previously mentioned, the solution methodologies that have been proposed in this case have been either heuristics or metaheuristics that involve the use of an approximation cost function to evaluate the solutions. In the case of Yang et al. (2000) two heuristics were proposed for the VRPSD with PR trips.

The numerical tests performed in Bianchi (2006) show that, when designing solution approaches for the VRPSD with PR trips being included as possible recourse actions, it is clearly preferable to approximate the cost of solutions when the available solution time for the problem is restricted. Good results are obtainable even when the approximation used is based on a function that does not explicitly consider the recourse cost. It was further observed in Rei et al. (2010) that, when solving the VRPSD under the classical recourse policy, with the exception of extreme cases where failures are observed at each customer along a route, the a priori routing cost of the optimal solution clearly outweighs the recourse cost (e.g., the relative weight of the recourse cost being approximately 5% of the total cost for a subset of instances that were described as challenging to solve, see Rei et al. (2010)). Therefore, when assessing the overall effort needed to solve the VRPSD, an important part of this effort should be devoted to finding good a priori routes. This being said, the stochastic nature of the problem cannot be simply ignored (i.e., the recourse cost remains appreciable). This is especially true in a context where VRP consistency is promoted by repeatedly applying the same a priori solution and, consequently, incurring the recourse cost each time the solution is used. Hence, there is a need to develop numerically efficient approximation functions for the recourse cost.

The general rule-based recourse approach that is proposed also serves this methodological purpose. Any ruled-based recourse, specified on a particular set of fixed rules, defines an upper bound on the recourse cost associated to the optimal restocking policy. Therefore, it can be used as a proxy to evaluate the cost of the a priori solutions in an overall solution process for the VRPSD. In the present paper, we will show that it can be effectively used to develop an efficient exact algorithm for the VRPSD.

3 A Rule-Based Recourse A Priori Model for the VRPSD

This section is dedicated to the presentation of the overall formulation applied to the VRPSD. Therefore, we first recall the a priori model that is used (Subsection 3.1). We then detail the recourse function defined to measure the expected routing costs involved in performing both the BF and PR trips in the second-stage following a fixed rule-based recourse policy. Thus, for a given a priori route and its policy, we show how the associated recourse cost can be efficiently computed using a recursive function (Subsection 3.2). Finally, we introduce a general class of volume-based recourse policies for the VRPSD (Subsection 3.3).

3.1 A Priori Model

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a complete undirected graph, where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ is the set of vertices and $\mathcal{E} = \{(v_i, v_j) | v_i, v_j \in \mathcal{V}, i < j\}$ is the set of edges. Vertex v_1 is the depot, where a fleet of m vehicles of capacity Q is based. Let vertex v_i ($i = 2, \ldots, n$) represent a customer whose demand ξ_i follows a discrete probability distribution with a finite support defined as $\{\xi_i^1, \xi_i^2, \ldots, \xi_i^l, \ldots, \xi_i^{s^i}\}$. We denote by p_i^l the probability that the l^{th} demand level (i.e., value ξ_i^l) occurs for ξ_i, i.e., $\mathbb{P}[\xi_i = \xi_i^l] = p_i^l$. Let c_{ij} denote the distance associated to edge (v_i, v_j).

As in Laporte et al. (2002), we assume that the expected demand of an a priori route does not exceed the vehicle capacity. The a priori model for the VRPSD can then be formulated as follows (we use here the original notation defined by Laporte et al. (2002)):

$$\underset{x}{\text{minimize}} \quad \sum_{i<j} c_{ij} x_{ij} + \mathcal{Q}(x) \tag{1}$$

$$\text{subject to} \quad \sum_{j=2}^{n} x_{1j} = 2m, \tag{2}$$

$$\sum_{i<k} x_{ik} + \sum_{k<j} x_{kj} = 2, \qquad k = 2, \ldots, n \tag{3}$$

$$\sum_{v_i, v_j \in S} x_{ij} \leq |S| - \left\lceil \frac{\sum_{v_i \in S} \mathbb{E}(\xi_i)}{Q} \right\rceil, \qquad (S \subset \mathcal{V} \setminus \{v_1\}; 2 \leq |S| \leq n-2) \tag{4}$$

$$0 \leq x_{ij} \leq 1, \qquad 2 \leq i < j < n \tag{5}$$

$$0 \leq x_{1j} \leq 2, \qquad j = 2, \ldots, n \tag{6}$$

$$x = (x_{ij}), \qquad \text{integer} \tag{7}$$

where,

$$\mathcal{Q}(x) = \sum_{k=1}^{m} \min\{\mathcal{Q}^{k,1}, \mathcal{Q}^{k,2}\}. \tag{8}$$

Function $\mathcal{Q}^{k,\rho}$ defines the expected recourse cost of the k^{th} vehicle-route when performed according to orientation ρ ($\rho = 1, 2$). As described in Dror and Trudeau (1986), the expected recourse cost of a route varies according to its orientation. Therefore, for each route in the a priori solution a specific orientation must be selected. As indicated in function (8), each route is evaluated using the two orientations and the one that minimizes the expected recourse cost is chosen. The specific computation of $\mathcal{Q}^{k,\rho}$ will be the subject of Subsection 3.2.

As for the overall formulation, the objective function (1) is defined as the total expected distance traveled by the vehicles (i.e., the sum of the distance traveled in performing the a priori routes and the expected distance traveled in performing the recourse actions considered). Constraints (2) and (3) define the structure of the a priori routes: each route starts and ends at the depot and each customer must be visited once. Inequalities (4) are the subtour elimination constraints, which also guarantee that the total expected demand of each route does not exceed a vehicle's capacity. Finally, constraints (5), (6) and (7) impose the necessary bounds and integrality restrictions on the decision variables.

3.2 The Recourse Function

In this subsection, we present the recourse function that is used for the VRPSD. Considering the set of a priori routes R, let us first consider an a priori route $i \in R$ expressed as vector $\vec{v} = (v_1 = v_{i_1}, v_{i_2}, \ldots, v_{i_t}, v_{i_{t+1}} = v_1)$. In addition, let us define vector $\vec{\theta} = (\theta_{i_2}, \ldots, \theta_{i_t})$, where $0 \leq \theta_{i_j} \leq Q$ for $j = 2, \ldots, t$, as the rule-based recourse policy associated with route $\vec{v}$. The process by which policy $\vec{\theta}$ is obtained will be the subject of the next subsection. For now, we simply assume that such a policy is given. The values in $\vec{\theta}$ are the residual capacity thresholds that specify when a vehicle performing $\vec{v}$ should carry out a PR trip. Therefore, when the vehicle leaves a scheduled customer v_{i_j} in $\vec{v}$ (i.e., after serving its demand ξ_{i_j}), it will perform a PR trip if its residual capacity is strictly below value θ_{i_j}, as illustrated in Figure 1. Considering that v_{i_t} is the last visited customer on route $\vec{v}$, value θ_{i_t} is simply set to zero. A numerical example of a threshold-based policy for route $\vec{v}$ is provided in Figure 1. In addition, as shown in the figure, a **Daily log-trip sheet** can be used to efficiently implement and record the necessary recourse actions (both the BF and PR trips) by the driver performing route $\vec{v}$ and to note the total distance traveled by the vehicle (i.e., the **Mileage** entry). When the vehicle performing $\vec{v}$ arrives at a customer v_{i_j} with a residual capacity of q, there are three mutually exclusive cases that can be observed. *First*, the demand realization of ξ_{i_j} exceeds value q (i.e., $q - \xi_{i_j} < 0$), which implies that a route failure occurs at v_{i_j}. In this case, the vehicle completes the service at the customer, via a split delivery, by performing a BF trip. It should be noted that this first case is independent of the threshold value of the considered customer (i.e., θ_{i_j}). *Second*, the demand realization of ξ_{i_j} does not exceed value q but $0 \leq q - \xi_{i_j} < \theta_{i_j}$. In this case, when $q - \xi_{i_j} = 0$, an exact stockout is observed, thus requiring a reactive recourse action (i.e., a BF trip). However, given the specific nature of this failure, the observed demand can still be served completely upon the arrival of the vehicle at the customer's location (i.e., a split delivery is not necessary). Therefore, following the return to the depot to restock, the vehicle proceeds to the next customer along the route (i.e., $v_{i_{j+1}}$). When $0 < q - \xi_{i_j} < \theta_{i_j}$, no failure is observed. However, the residual capacity of the vehicle, upon completion of the service of ξ_{i_j}, falls below the threshold value θ_{i_j}. Thus, a PR trip is performed and the route is resumed. *Third*, the demand realization of ξ_{i_j} does not exceed q and the difference between the two values is greater than θ_{i_j} (i.e., $q - \xi_{i_j} \geq \theta_{i_j}$). In this case, once the service of the demand is done, the vehicle directly proceeds to the next customer along the route (i.e., $v_{i_{j+1}}$).

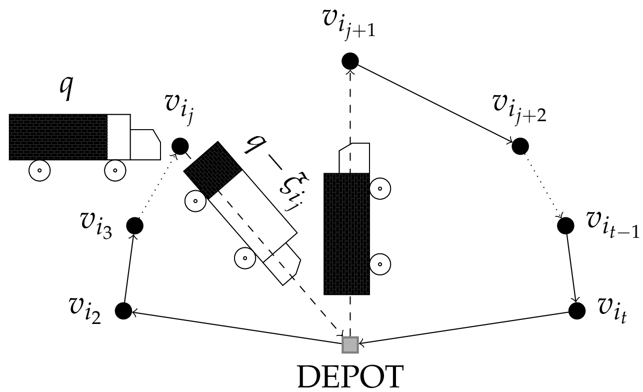
It should be noted that, whenever a route failure occurs the overall service at the customer is split. In turn, this entails that the loading/unloading process is duplicated and additional delays (e.g., stemming from the BF trips and the interruption of the service) are observed. It is assumed that such disruptions at a customer location generate an additional cost. This cost is defined as value b, and was also assumed by Yang et al. (2000).

We now develop the recourse function that is used in model (1)-(7). For a given route $\vec{v}$ and its associated policy $\vec{\theta}$, let us first define function $F_{i_j}(q)$ as the expected recourse cost of completing route $\vec{v}$ starting from vertex v_{i_j} (for $j = 1, \dots, t+1$) assuming that the vehicle arrives at the customer's location with a residual capacity of q (where $\theta_{i_{j-1}} \leq q \leq Q$). In view of the three cases previously described, function $F_{i_j}(q)$ is computed by applying the following recursive equation:

$$F_{i_j}(q) = \begin{cases} F_{i_j+1}(q) & \text{if } j = 1 \\ \mathbb{P}[q - \xi_{i_j} < 0] \left(b + 2c_{1i_j} + F_{i_j+1}(Q + q - \xi_{i_j}) \right) + & \\ \mathbb{P}[0 \leq q - \xi_{i_j} < \theta_{i_j}] \left(c_{1i_j} + c_{1i_j+1} - c_{i_j i_j+1} + F_{i_j+1}(Q) \right) + & \\ \mathbb{P}[q - \xi_{i_j} \geq \theta_{i_j}] F_{i_j+1}(q - \xi_{i_j}) & \text{if } j = 2, \dots, t \\ 0 & \text{if } j = t+1. \end{cases} \tag{9}$$

Given equation (9) and assuming that the k^{th} vehicle performs route $\vec{v}$, the expected recourse cost of the route can now be computed for the first orientation (i.e., $\rho = 1$) as follows:

$$\mathcal{Q}^{k,1} = F_{i_1}(Q). \tag{10}$$



(a)

Daily log-trip sheet

Cust. #	**Thresh**	**Rec. Act.**
DEPOT	–	
v_{i_2}	5	
v_{i_3}	10	
⋮	⋮	
v_{i_j}	θ_{i_j}	PR trip
$v_{i_{j+1}}$	$\theta_{i_{j+1}}$	
⋮	⋮	
$v_{i_{t-1}}$	8	
v_{i_t}	0	
DEPOT	–	
Mileage		

(b)

Figure 1: The vehicle is executing the a priori route $\vec{v} = (v_1 = v_{i_1}, v_{i_2}, \dots, v_{i_t}, v_{i_{t+1}} = v_1)$; the vehicle carries q units upon arrival at customer v_{i_j}; the residual capacity after demand realization satisfies the condition $0 \leq q - \xi_{i_j} < \theta_{i_j}$. Then, with a policy-based recourse the driver performs a preventive restocking trip.

Finally, to evaluate the expected recourse cost of the route for the second orientation (i.e., $\mathcal{Q}^{k,2}$), one simply needs to reverse the order of the vertices of $\vec{v}$ and reapply function (10).

3.3 Volume Based Recourse Policies for the VRPSD

In Subsection 3.2, we presented how the recourse function can be efficiently computed using the recursive equation (9). However, to evaluate (10) for a given route $\vec{v}$, one first needs to determine its associated rule-based recourse policy $\vec{\theta}$. Therefore, we now describe how such policies can be derived on the basis of a set of fixed operational rules that are prescribed by the company tasked with solving the VRPSD. In particular we consider a family of three volume-based policies.

Volume-based policies define the thresholds as a function of the demands of the customers or the capacity of the vehicles performing the routes. For a given route, such policies can implement straightforward operational rules that set the thresholds as a percentage of either the capacity of the vehicle, or, estimates obtained for the demands of the customers scheduled on the route. Given an a priori route i defined as $\vec{v} = (v_1 = v_{i_1}, v_{i_2}, \dots, v_{i_t}, v_{i_{t+1}} = v_1)$, three such policies are proposed. Let functions $\pi_p = \vec{v} \rightarrow \vec{\theta}$ (for $p = 1, 2, 3$), define them. The first policy π_1 applies the following operational rule: PR trips occur whenever the residual capacity of the vehicle performing the route falls below a preset percentage $\delta \in [0, 1]$ of its total capacity Q. In this case, the thresholds are all set to the same value: $\pi_1(\vec{v}) = (\theta_{i_2} = \delta Q, \dots, \theta_{i_j} = \delta Q, \dots, \theta_{i_t} = 0)$. This policy has the advantage of being straightforward to implement and allows an organization to easily adjust the operational rule to either be more conservative (i.e., higher values of δ, which tend to increase the number of PR trips performed) or less so (i.e., lower values of δ, which tend to decrease the number of PR trips performed).

In contrast with π_1, policies π_2 and π_3 tailor the threshold values according to the customers scheduled on a route. This is done by first generating point estimates for the demands. In the present case, the point estimates considered are the expected demand values: $\mathbb{E}(\xi_i)$, for $i = 1, \dots, n$. This being said, any demand estimates can be used to define π_2 and π_3. The second policy π_2 then applies the following operational rule: when leaving a customer v_{i_j}, that is scheduled on route $\vec{v}$, a PR trip is performed if the residual capacity of the vehicle is less than $\eta \mathbb{E}(\xi_{i_{j+1}})$, where $\eta \in \left[0, \dfrac{Q}{\mathbb{E}(\xi_{i_{j+1}})}\right]$. Therefore, the threshold value for a specific customer is set according to the demand estimate of the customer that immediately follows him in the sequence specified by the route $\vec{v}$: $\pi_2(\vec{v}) = (\theta_{i_2} = \eta \mathbb{E}(\xi_{i_3}), \dots, \theta_{i_j} = \eta \mathbb{E}(\xi_{i_{j+1}}), \dots, \theta_{i_t} = 0)$. As it is stated, policy π_2 computes the thresholds by applying a preset value η for all customers. However, this need not be the case and different values can also be applied to further tailor the thresholds for the customers. For example, based on the available information regarding the distributions of the demands, a company may adjust its operational rule by doing the following: increase the value η for a customer whose demand variance is high (i.e., thus being more conservative with respect to its recourse actions) and perform the reverse for a case where the variance is low (i.e., thus being less conservative with respect to the recourse actions). In an effort to simplify the analysis of the proposed policies, a single value will be used to perform the numerical experiments in Section 5.

Finally, the third policy π_3 applies the following operational rule: when leaving a customer v_{i_j}, that is scheduled on route $\vec{v}$, a PR trip is performed if the residual capacity of the vehicle is less than $\lambda \sum_{r=i_{j+1}}^{i_t} \mathbb{E}(\xi_r)$, where $\lambda \in \left[0, \frac{Q}{\sum_{r=i_{j+1}}^{i_t} \mathbb{E}(\xi_r)}\right]$. Similar to π_2, demand estimates are again used to compute π_3. However, the demand estimates of all remaining customers along the route are used here to define the value of a specific threshold: $\pi_3(\vec{v}) = (\theta_{i_2} = \lambda \sum_{r=i_3}^{i_t} \mathbb{E}(\xi_r), \ldots, \theta_{i_j} = \lambda \sum_{r=i_{j+1}}^{i_t} \mathbb{E}(\xi_r), \ldots, \theta_{i_t} = 0)$. Once more, it should be noted that a single fixed preset value λ is used to define π_3. However, different values can again be used in the operational rule, in this case, such values need to be set according to the subsequences of customers scheduled in $\vec{v}$. As previously stated, a single value will be applied here to simplify the numerical analysis of the policies.

4 The Solution Method

To solve model (1)-(7), defined under policies π_1, π_2 and π_3, we apply the Integer L-shaped algorithm, which has been shown to efficiently solve the VRPSD under the classical recourse policy (see Gendreau et al. (1995), Laporte et al. (2002) and Jabali et al. (2014)). This algorithm, which is based on the branch-and-cut paradigm, applies an exhaustive search of the first-stage decisional space while generating cuts that either enforce first-stage feasibility requirements to obtain the a priori routes (i.e., subtour elimination and capacity constraints), or, provide a lower bound on the recourse cost for both feasible and partial routes through the use of lower bounding functional (LBF) cuts. In order to present how this solution approach applies to the present model, we recall the general principles of the Integer L-shaped algorithm (Subsection 4.1), and the definition of partial routes and the lower bounding functional cuts (Subsection 4.2). We then develop lower bounding strategies that enable the application of the LBF cuts for the present problem (Section 4.3).

4.1 The Integer L-shaped Algorithm

Model (1)-(7) cannot be efficiently solved directly given the extremely large number of constraints involved in eliminating all possible subtours from the considered feasible set of routes and enforcing the capacity restrictions imposed (i.e., constraint set (4)). We recall that the computation of the recourse cost for a given route was discussed in section 3.2. To efficiently solve the model, the Integer L-shaped algorithm, which was originally proposed by Laporte and Louveaux (1993), applies a branch-and-cut strategy. This strategy entails the relaxation of the integrality constraints imposed on the decision variables, the subtour elimination and capacity restrictions, and the replacement of the recourse cost $\mathcal{Q}(x)$ by a valid lower bound Θ. Therefore, at a given iteration ν, the algorithm solves the following *current problem* (CP^ν):

$$CP^\nu : \min_{x,\Theta} \quad \sum_{i<j} c_{ij}x_{ij} + \Theta \tag{11}$$

subject to (2), (3), (5), (6),

$$\sum_{v_i,v_j \in S^k} x_{ij} \leq |S^k| - \left\lceil \frac{\sum_{v_i \in S^k} \mathbb{E}(\xi_i)}{Q} \right\rceil \qquad \forall k \in \mathbf{ST}^{\nu-1}, S^k \subset \mathcal{V} \setminus \{v_1\}, 2 \leq |S^k| \leq n-2, \tag{12}$$

$$L + (\Theta_p^q - L)\left(\sum_{h \in \mathbf{PR}^q} W_p^h(x) - |\mathbf{PR}^q| + 1 \right) \leq \Theta \quad \forall q \in \mathbf{PS}^{\nu-1}, p \in \{\alpha, \beta, \gamma\}, \tag{13}$$

$$L \leq \Theta \tag{14}$$

$$\sum_{\substack{1 \leq i \leq j \\ x_{ij}^m = 1}} x_{ij} \leq \sum_{1 \leq i \leq j} x_{ij}^f - 1 \qquad \forall f \in \mathbf{OC}^{\nu-1}. \tag{15}$$

Let (x^ν, Θ^ν) define the solution obtained for CP^ν. The first-stage solution x^ν is feasible for the original constraint sets (2), (3), (5) and (6). Thus, each route starts and ends at the depot, each customer is visited once and the necessary bounds are imposed on the first-stage variables. Let $\mathbf{ST}^{\nu-1}$ be an index set for all the subsets of vertices previously identified (i.e., throughout the first $\nu - 1$ iterations of the algorithm) and used to produce the cuts in (12). Thus, the routes defined by x^ν are also feasible for a subset of subtour elimination or capacity constraints, which are included in the cut set (12).

As for value Θ^ν, it defines a lower bound associated with the current first-stage solution x^ν (which may or may not be feasible). Value Θ^ν is determined according to the LBF cuts that have been added to CP^ν, constraints (13), and a general lower bound L that is valid over all feasible first-stage solutions, constraint (14). As will be detailed in Sections 4.2, the LBF cuts are defined according to general partial routes identified in partial solutions. We define $\mathbf{PS}^{\nu-1}$ as an index set for the partial solutions identified in the first $\nu-1$ iterations of the algorithm. Furthermore, for a given partial solution $q \in \mathbf{PS}^{\nu-1}$, let $h \in \mathbf{PR}^q$ be the set of partial routes contained in solution q (see Section 4.2). Lastly, we consider three topologies $p \in \{\alpha, \beta, \gamma\}$ for a general partial route, each yielding a valid lower bound Θ_p^q for all first-stage solutions.

Finally, constraint set (15) includes identified optimality cuts. Set $\mathbf{OC}^{\nu-1}$ includes an index for each feasible first-stage solution identified in the first $\nu-1$ iterations. Therefore, for each $f \in \mathbf{OC}^{\nu-1}$, a cut of type (15) is included in CP^ν to eliminate the feasible solution from further consideration.

The cut identification strategy applied at iteration ν then proceeds by first attempting to find violated subtour elimination and capacity constraints in solution x^ν. This is done by applying the separation heuristic procedures developed by Lysgaard et al. (2004) to identify these violated constraints. If such a constraint is identified, it is then added to the current problem and $\mathbf{ST}^\nu = \mathbf{ST}^{\nu-1} \cup \{k'\}$, where k' is the index associated with the subset of vertices defining the cut. In addition, a search for violated LBF cuts is also performed on solution x^ν. To do so, the exact separation procedure developed by Jabali et al. (2014) is applied to first search for general partial routes present in x^ν. Let $h' \in \mathbf{PR}^\nu$ be the general partial routes identified. A violated LBF cut is then obtained for $p \in \{\alpha, \beta, \gamma\}$ whenever $\Theta_p^\nu > \Theta^\nu$. In such a case, the cut is added to the current problem and $\mathbf{PS}^\nu$ is updated accordingly. When all of these separation procedures fail to identify violated cuts, a feasibility test is applied on solution x^ν. If the current solution is feasible, let f' be its associated index, an optimality cut is then added to the current problem and $\mathbf{OC}^\nu = \mathbf{OC}^{\nu-1} \cup \{f'\}$. Finally, the Integer L-shaped algorithm embeds this cut identification strategy in a branching procedure that terminates when optimality is established (see Jabali et al. (2014) for further details).

4.2 Lower Bounding Functionals

The LBFs (13) are generated based on general partial routes. These were initially proposed by Hjorring and Holt (1999) for the single-VRPSD, where a partial route was defined by a set of sequenced customers connected to a set of unsequenced customers that is connected to a set of sequenced customers. This structure was employed for the multi-VRPSD by Laporte et al. (2002). The concept of partial routes was further elaborated by Jabali et al. (2014), who treated partial routes as an alternating succession of sequenced sets and non-sequenced sets of customers. According to this definition, three topologies of LBFs were identified, one of which corresponds to the initial partial route defined by Hjorring and Holt (1999). In this paper, we employ the LBFs proposed by Jabali et al. (2014). In what follows, we define the LBFs using the notation proposed by Jabali et al. (2014), we then present the bounds used for the VRPSD under the policies π_1, π_2 and π_3.

General partial routes are identified based on partial solutions (i.e., solutions which do not yet include m feasible routes) of the CP^{ν}, solution x^{ν}. An illustration of a general partial route can be found in Figure (2), where the depot is duplicated for convenience. Let $\bar{\mathcal{G}}^{\nu}$ be the graph induced by the nonzero variables of the solution to CP^{ν}. A general partial route includes two types of components: 1) *Chains*, whose vertex sets are called chain vertex sets (CVSs), in which the vertices of a chain are connected to each other by edges (v_i, v_j), i.e., $x^{\nu}_{ij} = 1$ in $\bar{\mathcal{G}}^{\nu}$; 2) *Unstructured components*, whose vertex set are called unstructured vertex sets (UVSs). A chain is connected to a UVS by an *articulation vertex*. As previously mentioned, the exact separation procedure proposed by Jabali et al. (2014) is used in this paper to detect such partial routes. For $h \in \mathbf{PR}^{\nu}$, let κ denote the number of chains and

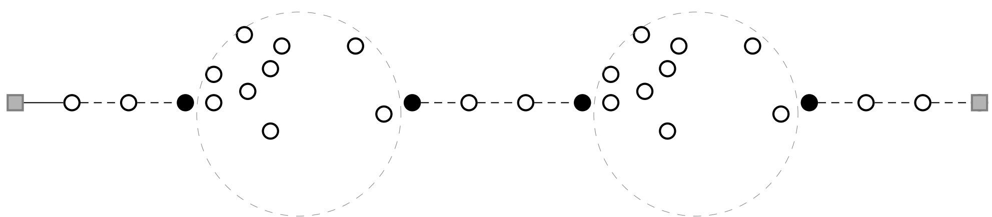


Figure 2: A general partial route h composed of sequenced and unsequenced sets.

$\kappa - 1$ denote the number of UVSs in partial route h. We denote by $S^t_h = \{v^t_{h_1}, \dots, v^t_{h_l}\}$ the t^{th} chain in partial route h, where $v^t_{h_k}$ is the k^{th} vertex in S^t_h, and h_l is the number of vertices in S^t_h. Therefore,

$$\sum_{(v_i,v_j)\in S^t_h} x^{\nu}_{ij} = |S^t_h| - 1, \ \forall t = 1, \dots, \kappa. \tag{16}$$

Let U^t_h be the t^{th} UVS in partial route h. Then,

$$\sum_{v_i,v_j\in U^t_h} x^{\nu}_{ij} = |U^t_h| - 1, \ \forall t = 1, \dots, \kappa - 1. \tag{17}$$

A UVS is preceded by a chain and proceeded by another. Therefore,

$$\sum_{v_j\in U^t_h} x^{\nu}_{h^t_l,j} = 1, \ \forall t \leq \kappa - 1, \tag{18}$$

and

$$\sum_{v_j\in U^{t-1}_h} x^{\nu}_{h^t_1,j} = 1, \ \forall t \geq 2 \tag{19}$$

The interest to generalize the structure of a partial route h is motivated by the fact that each chain may be viewed as a special case of a UVS, and each articulation vertex can be assumed as a single-CVS. Based on these observations, three partial route topologies were derived.

Figure (3a) shows an example of an α-route topology, where the first and last chains are viewed as CVSs, while the intermediate component containing multiple chains and UVSs is viewed as a single-UVS. This case corresponds to the partial route topology proposed by Hjorring and Holt (1999). Figure (3b) illustrates the case of a β-route topology, where the actual alternation of CVSs and UVSs is maintained. Figure (3c) shows an example of a γ-route topology, where each chain is viewed as a UVS and articulation vertices are viewed as single-CVSs.

We now present the definition of the functional $W_p^h(x)$, which is stated in equation (20), and recall its purpose in the LBF cuts, i.e., constraints (13). Finally, in Section 4.3 we develop lower bounding strategies to obtain the values Θ_p^q, tailored to the recourse cost defined according to policies π_1, π_2 and π_3.

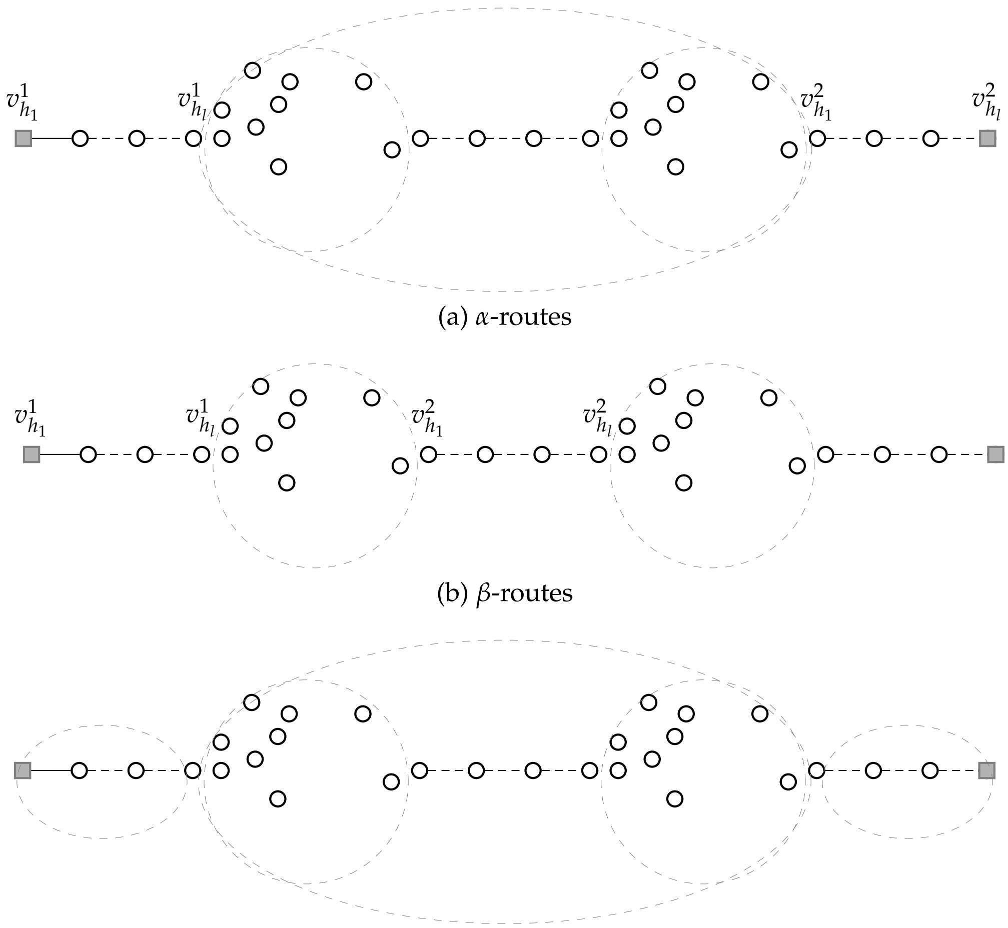
Given a general partial route h, the choice of a topology $p \in \{\alpha, \beta, \gamma\}$ defines the specific succession of CVSs and UVSs that are used to produce the LBF cut. Specifically, a topology fixes the vertices that are included in sets S_h^t, for $t = 1, \ldots, \kappa$, and U_h^t, for $t = 1, \ldots, \kappa - 1$. The functional $W_p^h(x)$, introduced by Jabali et al. (2014), is defined as follows,

$$
\begin{aligned}
W_p^h(x) = {} & \sum_{t=1}^{\kappa} \sum_{\substack{(v_i,v_j)\in S_h^t \\ v_i \neq v_1}} 3x_{ij} + \sum_{(v_1,v_j)\in S_h^1} x_{1j} + \sum_{(v_1,v_j)\in S_h^\kappa} x_{1j} + \sum_{t=1}^{\kappa-1} \sum_{v_i,v_j\in U_h^t} 3x_{ij} \\
& + \sum_{t=1}^{\kappa-1} \sum_{\substack{v_j\in U_h^t \\ v_{h_l}^t \neq v_1}} 3x_{h_l^t j} + \sum_{t=2}^{\kappa} \sum_{\substack{v_j\in U_h^{t-1} \\ v_{h_1}^t \neq v_1}} 3x_{h_1^t j} + \sum_{\substack{v_j\in U_h^1 \\ v_{h_l}^1 = v_1}} x_{h_l^1 j} + \sum_{\substack{v_j\in U_h^{b-1} \\ v_{h_1}^\kappa = v_1 \\ v_{h_1}^{\kappa-1} \neq v_1}} x_{h_1^\kappa j} \\
& - (3|R_h| - 5).
\end{aligned} \tag{20}
$$

We refer the reader to Jabali et al. (2014) for the proof of validity of equation (20) as a component of the LBF cut (13). We simply summarize that, for a given topology p, if a solution x follows the succession of CVSs and UVSs prescribed for the general partial route h, then $W_p^h(x) = 1$, otherwise $W_p^h(x) \leq 0$. Therefore, considering a partial solution q, $\sum_{h\in\mathbf{PR}^q} W_p^h(x) = |\mathbf{PR}^q|$ if and only if x follows the succession of CVSs and UVSs prescribed for all the partial routes included in $\mathbf{PR}^q$. This entails that $\Theta_p^q \leq \Theta$.

4.3 Bounding the Recourse Cost

Considering a specific partial solution q that includes a partial route $h \in \mathbf{PR}^q$, in the present section, we describe the computation of Θ_p^{qh}, which is the lower bound associated to h



(a) α-routes

(b) β-routes

(c) γ-routes

Figure 3: Partial route topologies.

when topology $p \in \{\alpha, \beta, \gamma\}$ is applied to generate an LBF cut (13). Moreover, the bound Θ^q_p, which is included in (13), is fixed to the sum of the lower bounds associated with the different partial routes associated with q, i.e., $\Theta^q_p = \sum\limits_{h \in \mathbf{PR}^q} \Theta^{qh}_p$. In the following, to alleviate the notation, we will drop the index q and simply refer to the lower bound Θ^h_p (i.e., a partial route is always associated with a partial solution). Furthermore, we focus on deriving value Θ^h_α (i.e., the specific topology $p = \alpha$). This is motivated by the fact that the computation of Θ^h_α can be easily generalized to evaluate both Θ^h_β and Θ^h_γ, considering that topologies β and γ can be viewed as containing successive α-route structures. We next present the strategy to compute Θ^h_α under the first two policies (i.e., π_1 and π_2), which can be done in a unified way. We then conclude the present subsection by detailing the specificities of evaluating Θ^h_α when the third policy is applied (i.e., π_3).

Bounding the Policies π_1 and π_2

Let h be a partial route that is assumed to follow topology α. We denote the ordered vertex sets in chain S_h^1 and S_h^2 as $\{v_{h_1}^1, \ldots, v_{|S_h^1|}^1\}$ and $\{v_{h_1}^2, \ldots, v_{|S_h^2|}^2\}$, respectively. We recall that in topology α there is a single UVS, i.e., U_h^1. Partial route h can then be represented as follows $(v_1 = v_{h_1}^1, \ldots, v_{|S_h^1|}^1, U_h^1, v_{h_1}^2, \ldots, v_{|S_h^2|}^2 = v_1)$. Let $l = |U_h^1|$, for the sake of simplifying the subsequent recursion formulas, we redefine the partial route h, in similar terms as route i, as follows

$$h = (v_1 = v_{i_1}, \ldots, v_{i_{j-l}}, \{v_{i_{u_1}}, v_{i_{u_2}}, \ldots, v_{i_{u_l}}\}, v_{i_{j+1}}, \ldots, v_{i_{t+1}} = v_1),$$

where the articulation vertices $v_{|S_h^1|}^1$ and $v_{|S_h^2|}^2$ are now denoted by $v_{i_{j-l}}$ and $v_{i_{j+1}}$, respectively. Using partial route h, we define an artificial route $\tilde{h}$ as follows,

$$\tilde{h} = (v_1 = v_{i_1}, \ldots, v_{i_{j-l}}, \square_{i_{j-l+1}}, \square_{i_{j-l+2}}, \ldots, \square_{i_j}, v_{i_{j+1}}, \ldots, v_{i_{t+1}} = v_1), \tag{21}$$

where each possible ordering of the l unsequenced customers included in U_h^1 can be assigned to the positions $\square_{i_{j-l+1}}, \ldots, \square_{i_j}$. In what follows, we refer to $\square_{i_j}$ as the j^{th} position in artificial route $\tilde{h}$, and we develop a bounding procedure for $\tilde{h}$ which essentially bounds positions $\square_{i_{j-l+1}}, \ldots, \square_{i_j}$.

To introduce the notation used to derive the proposed lower bounding procedure, let us recall that function $F_{i_j}(.)$, as previously defined in (9), provides the exact computation of the expected recourse cost onward from the j^{th} customer when both customers j^{th} and $j+1^{\text{th}}$ are known, e.g., for two consecutive customers in a chain. In what follows, we primarily reconstruct recursive formula (9) in a manner that yields a bound on the unsequenced customers in U_h^1. Let $\widetilde{F}_{i_j}(.)$ represent an absolute lower bound for the expected recourse cost of the j^{th} position of artificial route $\tilde{h}$. Let $\hat{F}_{i_j}(.)|_{i_j := u_e}$ be the lower bound for a specific unsequenced customer $v_{u_e} \in U_h^1$ that would be assigned to the j^{th} position of the artificial route $\tilde{h}$.

Considering a sequenced route, we introduce a bounding structure in Lemma 4.1 for $\hat{F}_{i_k}(.)|_{i_k := u_e}$, which is constructed based on the knowledge of the absolute bounds on customer k, i.e. $\widetilde{F}_{i_k}(.)$, for $k > j$. We then develop the bounding structure proposed in Lemma 4.1 to bound artificial route $\tilde{h}$. This is done in two main steps, in Lemma 4.2 an absolute lower bound on the expected recourse cost for the j^{th} position in the artificial route is established. This is then recursively embedded in Lemma 4.3 to obtain bounds for positions $j - l + 1 \leq k < j$ in artificial route $\tilde{h}$.

We begin by showing how a valid lower bound can be computed for a feasible route $\vec{v} = (v_1 = v_{i_1}, v_{i_2}, \ldots, v_{i_k}, v_{i_{k+1}}, \ldots, v_{i_t}, v_{i_{t+1}} = v_1)$ under policies π_1 and π_2. We recall that $\pi_1(\vec{v}) = (\theta_{i_2} = \delta Q, \ldots, \theta_{i_j} = \delta Q, \ldots, \theta_{i_t} = 0)$ and $\pi_2(\vec{v}) = (\theta_{i_2} = \eta\mathbb{E}(\xi_{i_3}), \ldots, \theta_{i_j} = \eta\mathbb{E}(\xi_{i_{j+1}}), \ldots, \theta_{i_t} = 0)$. By defining the minimum and maximum threshold values of the route $\vec{v}$ as $\underline{\theta}_{\vec{v}} = \min\{\theta_{i_2}, \ldots, \theta_{i_k}, \theta_{i_{k+1}}, \ldots, \theta_{i_{t-1}}\}$ and $\overline{\theta}_{\vec{v}} = \max\{\theta_{i_2}, \ldots, \theta_{i_k}, \theta_{i_{k+1}}, \ldots, \theta_{i_{t-1}}\}$, respectively, then the following result stands.

Lemma 4.1. *Let q denote the residual capacity of the vehicle upon arriving at v_{i_k}. Let*

$$\hat{F}_{i_k}(q) = \begin{cases} \widetilde{F}_{i_{k+1}}(q) & \text{if } k = 1 \\ \mathbb{P}[q - \xi_{i_k} < 0]\left(b + 2c_{1i_k} + \widetilde{F}_{i_{k+1}}(Q + q - \xi_{i_k})\right) + & \\ \mathbb{P}[0 \leq q - \xi_{i_k} < \underline{\theta}_{\vec{v}}]\left(\tilde{c}_{i_k} + \widetilde{F}_{i_{k+1}}(Q)\right) + & \\ \mathbb{P}[q - \xi_{i_k} \geq \overline{\theta}_{\vec{v}}]\widetilde{F}_{i_{k+1}}(q - \xi_{i_k}) & \text{if } k = 2, \ldots, t \\ 0 & \text{if } k = t+1, \end{cases} \tag{22}$$

where $\tilde{c}_{i_k} = \min_{a=k+1,\ldots,t}\{c_{1,i_k} + c_{1,i_a} - c_{i_k,i_a}\}$ and $\widetilde{F}_{i_{k+1}}(.) \leq F_{i_{k+1}}(.)$, then $\hat{F}_{i_k}(q) \leq F_{i_k}(q)$ for all q.

Proof. We recall $F_{i_k}(q)$ from (9) as

$$F_{i_k}(q) = \begin{cases} F_{i_{k+1}}(q) & \text{if } k = 1 \\ \mathbb{P}[q - \xi_{i_k} < 0]\left(b + 2c_{1i_k} + F_{i_{k+1}}(Q + q - \xi_{i_k})\right) + & \\ \mathbb{P}[0 \leq q - \xi_{i_k} < \theta_{i_k}]\left(c_{1i_k} + c_{1i_{k+1}} - c_{i_k i_{k+1}} + F_{i_{k+1}}(Q)\right) + & \\ \mathbb{P}[q - \xi_{i_k} \geq \theta_{i_k}]F_{i_{k+1}}(q - \xi_{i_k}) & \text{if } k = 2, \ldots, t \\ 0 & \text{if } k = t+1. \end{cases}$$

Since each term in $\hat{F}_{i_k}(q)$ is a direct lower bound value for its counterpart term in the $F_{i_k}(q)$ then $\hat{F}_{i_k}(q) \leq F_{i_k}(q)$. □

It should first be noted that $\tilde{h}$ includes two sequenced parts (i.e., chains S_h^1 and S_h^2). Therefore, for all possible values q, the onward expected recourse cost after the j^{th} position can be computed exactly using (9) (i.e., $\widetilde{F}_{i_k}(q) = F_{i_k}(q)$ for $j < k \leq t+1$). We now present a lower bound on the onward recourse cost for the j^{th} position in $\tilde{h}$.

Lemma 4.2. *A lower bound on the expected recourse cost for the j^{th} position in the artificial route $\tilde{h}$ can be defined as follows:*

$$\widetilde{F}_{i_j}(q) = \min_{v_{u_e} \in U_h^1} F_{i_j}(q)|_{i_j := u_e} \; \forall q \tag{23}$$

where $F_{i_j}(q)|_{i_j := u_e}$ is computed by assigning $v_{u_e} \in U_h^1$ at the j^{th} position in $\tilde{h}$, and then applying the recourse function (9).

Proof. Since the j^{th} position is unsequenced in $\tilde{h}$, and considering that it can potentially be assigned to each $v_{u_e} \in U_h^1$, a valid lower bound for the onward expected recourse cost at the j^{th} position is obtained by minimizing the recourse cost over U_h^1 for each q. Then, $\widetilde{F}_{i_j}(.) \leq F_{i_j}(.)|_{i_j:=u_e}$ is implied by the definitions. □

By embedding Lemma 4.2 within Lemma 4.1, a valid lower bound can be derived for the positions not yet sequenced in $\tilde{h}$, i.e., $(\square_{i_{j-l+1}}, \square_{i_{j-l+2}}, \dots, \square_{i_{j-1}})$. Therefore, at the $j-1^{\text{th}}$ position, Lemma 4.2 is used to obtain a lower bound for each $v_{u_e} \in U_h^1$. This process is then sequentially applied to bound the remaining positions.

Lemma 4.3. *A lower bound for the expected recourse cost at k^{th} position of artificial route $\tilde{h}$ for $j-l+1 \leq k < j$ can be computed as follows:*

$$\widetilde{F}_{i_k}(q) = \min_{v_{u_e} \in U_h^1} \hat{F}_{i_k}(q)|_{i_k:=u_e} \; \forall q, \tag{24}$$

in which $\hat{F}_{i_k}(q)|_{i_k:=u_e}$ is defined as

$$\hat{F}_{i_k}(q)|_{i_k:=u_e} = \begin{cases} \mathbb{P}[q - \xi_{u_e} < 0] \left(b + 2c_{1u_e} + \widetilde{F}_{i_{k+1}}(Q + q - \xi_{u_e})\right) + \\ \mathbb{P}[0 \leq q - \xi_{u_e} < \underline{\theta}_{U_h^1}] \left(\tilde{c}_{u_e} + \widetilde{F}_{i_{k+1}}(Q)\right) + \\ \mathbb{P}[q - \xi_{u_e} \geq \overline{\theta}_{U_h^1}] \widetilde{F}_{i_{k+1}}(q - \xi_{u_e}) \end{cases} \tag{25}$$

where, $\underline{\theta}_{U_h^1} = \min_{v_{u_e} \in U_h^1} \theta_{u_e}$, $\overline{\theta}_{U_h^1} = \max_{v_{u_e} \in U_h^1} \theta_{u_e}$ and

$$\tilde{c}_{u_e} = \min_{v_{u_{e'}} \in U_h^1 : v_{u_{e'}} \neq v_{u_e}} \{c_{1,u_e} + c_{1u_{e'}} - c_{u_e,u_{e'}}\}$$

defines the minimum PR trip cost that can be done from v_{u_e} within U_h^1, given $\widetilde{F}_{i_{k+1}}(q), \dots, \widetilde{F}_{i_j}(q)$, $\forall q$.

Proof. Let us consider position i_{j-1}, where the valid lower bound $\widetilde{F}_{i_j}(.)$ is assumed known, considering Lemma 4.2. Let

$$\hat{F}_{i_{j-1}}(q)|_{i_{j-1}:=u_e} = \begin{cases} \mathbb{P}[q - \xi_{u_e} < 0] \left(b + 2c_{1u_e} + \widetilde{F}_{i_j}(Q + q - \xi_{u_e})\right) + \\ \mathbb{P}[0 \leq q - \xi_{u_e} < \underline{\theta}_{U_h^1}] \left(\tilde{c}_{u_e} + \widetilde{F}_{i_j}(Q)\right) + \\ \mathbb{P}[q - \xi_{u_e} \geq \overline{\theta}_{U_h^1}] \widetilde{F}_{i_j}(q - \xi_{u_e}) \end{cases}$$

define the intermediate lower bound for the onward expected recourse cost at position i_{j-1} if customer v_{u_e} is placed there (see Lemma 4.1). By defining $\widetilde{F}_{i_{j-1}}(q) = \min_{v_{u_e} \in U_h^1} \hat{F}_{i_{j-1}}(q)|_{i_{j-1}:=u_e}$, value $\widetilde{F}_{i_{j-1}}(q)$ clearly defines a lower bound for $F_{i_{j-1}}(q)$. Furthermore, this result holds for all positions k, where $j-l+1 \leq k < j-1$. □

For the ${i_{j-l}}^{\text{th}}$ customer (i.e., articulation vertex in S_h^1), a lower bound for the expected recourse cost can be computed as follows:

$$\widetilde{F}_{i_{j-l}}(q) = \hat{F}_{i_{j-l}}(q) \ \forall q,$$

where

$$\hat{F}_{i_{j-l}}(q) = \begin{cases} \mathbb{P}[q - \xi_{i_{j-l}} < 0] \left(b + 2c_{1i_{j-l}} + \widetilde{F}_{i_{j-l+1}}(Q + q - \xi_{i_{j-l}})\right) + \\ \mathbb{P}[0 \leq q - \xi_{i_{j-l}} < \underline{\theta}_{U_h^1}] \left(\tilde{c}_{i_{j-l}} + \widetilde{F}_{i_{j-l+1}}(Q)\right) + \\ \mathbb{P}[q - \xi_{i_{j-l}} \geq \overline{\theta}_{U_h^1}] \widetilde{F}_{i_{j-l+1}}(q - \xi_{i_{j-l}}) \end{cases} \tag{26}$$

given that $\widetilde{F}_{i_{j-l+1}}(q)$ for all q is computed using Lemma 4.3 and where $\tilde{c}_{i_{j-l}} = \min\limits_{v_{u_e} \in U_r} \{c_{1,i_{j-l}} + c_{1,u_e} - c_{i_{j-l},u_e}\}$ defines the minimum PR trip cost that could be incurred from $v_{i_{j-l}}$ into U_h^1.

Finally, for the remaining portion of the artificial route $\tilde{h}$, i.e., $v_{i_1}, \ldots, v_{i_{j-l-1}}$, we note that the recourse function (9) can be used to successively compute $F_{i_{j-l-1}}(.)$,..., $F_{i_1}(.)$ (i.e., $\widetilde{F}_{i_k}(q) = F_{i_k}(q)$ for $1 < k \leq j - l - 1$). Then $\widetilde{F}_{i_1}(Q) = \tilde{\mathcal{Q}}_{\tilde{h}}^{k,1}$, can be used to complete the computation of the lower bound value. As for obtaining value $\tilde{\mathcal{Q}}_{\tilde{h}}^{k,2}$, we simply reverse the artificial route and apply the same computation. Therefore, $\Theta_\alpha^h = \min\{\tilde{\mathcal{Q}}_{\tilde{h}}^{k,1}, \tilde{\mathcal{Q}}_{\tilde{h}}^{k,2}\}$ results in a lower bound value for recourse cost for the partial route h.

Bounding the Policy π_3

In the case of policy π_3, the computation of the recourse cost for the artificial route $\tilde{h}$ remains unchanged with the exception of the threshold values used (i.e., $\underline{\theta}_{U_h^1}$ and $\overline{\theta}_{U_h^1}$ in Lemma 4.3). These threshold values now need to be determined according to the specific positions associated with U_h^1. Let us define $\underline{\theta}_{U_h^1}^k$ and $\overline{\theta}_{U_h^1}^k$ as the aforementioned threshold values associated with position k, for $j - l + 1 \leq k < j$. To express these values, we define 1^{st}, 2^{nd}, ..., $l - 1^{\text{th}}$ minimum and maximum expected demands associated with the customers included in U_h^1 as follows,:

$$y_1 = \underset{v_{u_e} \in U_h^1}{} \mathbb{E}(\xi_{v_{u_e}}), y_2 = \underset{v_{u_e} \in U_h^1 \setminus \{v_{u_{y_1}}\}}{} \mathbb{E}(\xi_{v_{u_e}}), \ldots, y_{l-1} = \underset{v_{u_e} \in U_h^1 \setminus \{v_{u_{y_1}}, \ldots, v_{u_{y_{l-2}}}\}}{} \mathbb{E}(\xi_{v_{u_e}})$$

$$z_1 = \underset{v_{u_e} \in U_h^1}{} \mathbb{E}(\xi_{v_{u_e}}), z_2 = \underset{v_{u_e} \in U_h^1 \setminus \{v_{u_{z_1}}\}}{} \mathbb{E}(\xi_{v_{u_e}}), \ldots, z_{l-1} = \underset{v_{u_e} \in U_h^1 \setminus \{v_{u_{z_1}}, \ldots, v_{u_{z_{l-2}}}\}}{} \mathbb{E}(\xi_{v_{u_e}})$$

Let us recall that policy π_3 is defined as $\pi_3(\vec{v}) = (\theta_{i_2} = \lambda \sum_{r=i_3}^{i_t} \mathbb{E}(\xi_r), \ldots, \theta_{i_j} = \lambda \sum_{r=i_{j+1}}^{i_t} \mathbb{E}(\xi_r), \ldots, \theta_{i_t}$ 0) for a given route $\vec{v} = (v_1 = v_{i_1}, v_{i_2}, \ldots, v_{i_t}, v_{i_{t+1}} = v_1)$. Considering that the artificial route $\tilde{h} = (v_1 = v_{i_1}, \ldots, v_{i_{j-l}}, \square_{i_{j-l+1}}, \square_{i_{j-l+2}}, \ldots, \square_{i_j}, v_{i_{j+1}}, \ldots, v_{i_{t+1}} = v_1)$, is unsequenced from the $j - l + 1^{\text{th}}$ position up to the j^{th} position, we set values $\underline{\theta}_{U_h^1}^k$ and $\overline{\theta}_{U_h^1}^k$, for $j - l + 1 \leq k < j$ as follows

$$\underline{\theta}^{k}_{U^1_h} = \lambda\Big(\sum_{a=1}^{j-k}\mathbb{E}(\xi_{y_a}) + \sum_{r=i_{j+1}}^{i_t}\mathbb{E}(\xi_r)\Big),\ \overline{\theta}^{k}_{U^1_h} = \lambda\Big(\sum_{a=1}^{j-k}\mathbb{E}(\xi_{z_a}) + \sum_{r=i_{j+1}}^{i_t}\mathbb{E}(\xi_r)\Big).$$

Finally, to compute $\hat{F}_{i_{j-l}}(q)$ using (26), under policy π_3, values $\theta^{min}_{U^1_h}$ and $\theta^{max}_{U^1_h}$ are simply set to

$$\theta^{min}_{U^1_h} = \theta^{max}_{U^1_h} = \lambda\Big(\sum_{v_{ue}\in U^1_h}\mathbb{E}(\xi_{v_{ue}}) + \sum_{r=i_{j+1}}^{i_t}\mathbb{E}(\xi_r)\Big).$$

We have presented the computation of the bounds associated with Θ^h_α. This computation is generalized, to both Θ^h_β and Θ^h_γ, as these can be viewed as successive α-route structures.

5 Numerical Result

In this section, we present extensive computational experiments conducted to assess the effectiveness of the solution method, as well as the quality of the three rule-based recourses proposed. In the set of instances designed for these numerical experiments both customer locations and the demand distribution functions are randomly generated. In each instance, a set of n vertices including the depot and $n-1$ customers as $\{v_1,\ldots,v_n\}$ are scattered in a square of $[0,100]^2$ according to a continuous uniform distribution. For each pair v_i and v_j, the traveling cost c_{ij} is then set to the nearest integer associated to the Euclidean distance between the two vertices. It should also be noted that the cost value b is defined as the average distance to the depot when considering all customers (i.e., $b = \sum_{i=2,\ldots,n} c_{i1}/(n-1)$). As previously defined, b is incurred whenever a failure occurs when applying a route to represent the cost associated with the added disturbance from the customer's perspective of having its demand serviced on two consecutive visits. Such a cost can be adjusted to reflect the overall quality of service that a transportation company is interested in offering to its customers. As for the specific choice of the value b that is considered, the motivation was to ensure that it scales (i.e., defined on comparable units of measurement) to the overall costs used in the objective function of the VRPSD, which depends of the travel cost.

Three demand ranges $[1,5]$, $[6,10]$, and $[11,15]$ are selected to present low, medium, and high demand customers. Each customer $v_i \in \{v_2,\ldots,v_n\}$ is then assigned to one of these three ranges with equiprobability. Next, five demand realizations based on the assigned ranges are generated for each customer v_i and the probabilities $\{0.1,0.2,0.4,0.2,0.1\}$ are associated to each value within the specific interval. The filling coefficient and vehicle capacity are defined through the function $\bar{f} = \frac{\sum_{i=2}^{n}\mathbb{E}(\xi_i)}{mQ}$, where m is the number of homogeneous vehicles with capacity Q. Four filling coefficients $\bar{f} = 0.90, 0.92, 0.94,$ and 0.96 are used to compute Q, where $m = 2, 3,$ and 4. The computational study is performed on a set of 11 possible pairs of (n, m) as indicated in Table (1). For each pair, 10 instances are randomly generated (providing 110 base instances). Considering the four filling coefficients for each pair of (n, m), a total of 440 instances are thus generated.

Three volume rule-based policies are examined in this paper. As stated in §3.3, let us recall that policy π_1 is based on a preset percentage δ of the capacity of the vehicles, while policies π_2 and π_3 are defined according to fixed coefficients (i.e., η and λ for π_2 and π_3, respectively) applied to either the expected demand of the subsequent customer along the considered route (i.e., policy π_2), or, the total expected demands of the remaining customers sequenced on the considered route (i.e., policy π_3). It should be noted that these policies, more precisely their preset coefficients, need to be tuned and calibrated carefully by decision makers facing the problems. These threshold policies govern how return trips to the depot are performed and can be used to formulate varying levels of risk aversion from the decision maker's perspective. As an overall principle, by increasing the preset coefficients under the different policies, vehicles will perform PR trips more often and less failures are expected to be observed, while a reduction in the coefficient values would have the reverse effect (i.e., a higher risk of observing failures).

To perform a thorough numerical analysis, three preset values for each policy are selected: $\delta = 0.02, 0.03, 0.05$, $\eta = 0.80, 1.00, 1.25$, and $\lambda = 0.80, 0.90, 1.00$. These values where chosen to enable a proper calibration of the policies to be performed and to assess the impact of using different threshold levels. Therefore, for each considered policy, a median value was first selected: $\delta = 0.03$ for π_1, $\eta = 1.00$ for π_2 and $\lambda = 0.90$ for π_3, which defines the benchmark in each case. Two alternate values were then defined for each policy to represent a more risk averse operational rule set with respect to the occurrence of route failures (i.e., $\delta = 0.05$, $\eta = 1.25$ and $\lambda = 1.00$) and a less risk averse approach (i.e., $\delta = 0.02$, $\eta = 0.80$ and $\lambda = 0.80$). To summarize the numerical experiments conducted, each instance is solved under the three policies that are applied using each preset value, thus a total of $3,960$ runs are performed.

The Integer L-shaped algorithm was programmed in C++ using ILOG CPLEX 12.6. The subtour elimination and capacity constraints (4) are generated using the CVRPSEP package of Lysgaard et al. (2004) and the branching procedure, which is used for the L-shaped algorithm, is implemented using the OOBB package developed by Gendron et al. (2005). We use three topologies $p \in \{\alpha, \beta, \gamma\}$ for generating general partial route cuts. All experiments were conducted on a cluster of 27 machines each having two Intel(R) Xeon(R) X5675 3.07 GHz processors with 96 GB of RAM running on Linux. Each machine has 12 cores and each experiment was run using a single thread. An optimality gap of 0.01% was imposed as well as a maximum CPU run time of 10 hours on all runs. Therefore, if the algorithm reaches the maximum allotted time without finding a solution within the desired gap, the best integer feasible solution found is simply reported.

Table 1: Combinations of parameters to generate instances

n	m	$\bar{f}$
20	2	0.90, 0.92, 0.94, 0.96
30	2	0.90, 0.92, 0.94, 0.96
40	2, 3, 4	0.90, 0.92, 0.94, 0.96
50	2, 3, 4	0.90, 0.92, 0.94, 0.96
60	2, 3, 4	0.90, 0.92, 0.94, 0.96

The obtained results are analyzed in the next two subsections. In Subsection 5.1, the three proposed policies are evaluated in terms of the computational effort needed to solve the VRPSD when each of them is used to define the recourse cost. While in Subsection 5.2, a solution cost assessment is conducted for the proposed policies.

5.1 Computational Policy Analysis

The results obtained for all numerical experiments are summarized in Tables 2, 3, and 4, each table corresponds to the results of a single policy. These results are aggregated according to the pair (n, m) and the filling coefficient $\bar{f}$ defining the instances, as well as the preset values associated with the policies (i.e., δ, η and λ for π_1, π_2 and π_3, respectively). Results are reported as follows: 1) the "Solved" columns presents the number of instances (out of ten for each aggregated category) that were solved to optimality by the Integer L-shaped algorithm; 2) the "Time" columns refer to the average running times in seconds that were needed by the algorithm to solve those instances to optimality; 3) the "Gap" columns present the average optimality gap obtained by the algorithm over all instances solved (i.e., both those solve optimally and those for which only a feasible solution was obtained).

When analyzing the results in Tables 2, 3, and 4, one first observes the general trend that was previously reported by Gendreau et al. (1995), Laporte et al. (2002), and Jabali et al. (2014) regarding the overall complexity related to solving the VRPSD. Therefore, regardless of the specific policy used, the complexity of solving the problem tends to increase as the number of customers, number of vehicles, and the filling coefficients increase. This trend is illustrated via both the number of instances solved to optimality that tend to decrease as the values of the instances parameters (n, m) and $\bar{f}$ increase, and the running times which tend to increase as the value $\bar{f}$ increases for fixed values for the pair (n, m).

Next, we analyze how the algorithm performs when solving the VRPSD under the three rule-based policies proposed. As reported in Tables 2, 3, and 4, on a total of 1,320 runs (which were performed using each considered policy), the Integer L-shaped algorithm obtained optimal solutions in 655 runs using π_1, 683 runs using π_2 and 593 runs using π_3. From these results, it clearly appears that the Integer L-shaped algorithm is most efficient

Table 2: Result of running the fixed policy π_1.

n	m	δ	$\bar{f}$	Solved	Time(s)	Gap	$\bar{f}$	Solved	Time(s)	Gap	$\bar{f}$	Solved	Time(s)	Gap	$\bar{f}$	Solved	Time(s)	Gap
20	2	0.02	0.90	10	22.10	0.00%	0.92	10	15.50	0.00%	0.94	10	18.40	0.00%	0.96	9	705.56	0.11%
20	2	0.03	0.90	10	12.20	0.00%	0.92	10	13.50	0.00%	0.94	10	15.10	0.00%	0.96	9	503.11	0.08%
20	2	0.05	0.90	10	13.80	0.00%	0.92	10	14.00	0.00%	0.94	10	15.40	0.00%	0.96	9	539.33	0.08%
30	2	0.02	0.90	10	20.20	0.00%	0.92	9	481.22	0.04%	0.94	10	2405.00	0.00%	0.96	9	2771.22	0.16%
30	2	0.03	0.90	10	23.20	0.00%	0.92	9	407.56	0.13%	0.94	10	5407.10	0.00%	0.96	9	2432.44	0.16%
30	2	0.05	0.90	10	17.90	0.00%	0.92	9	412.89	0.16%	0.94	9	2415.22	0.07%	0.96	7	5183.57	0.40%
40	2	0.02	0.90	10	21.70	0.00%	0.92	10	98.80	0.00%	0.94	10	61.80	0.00%	0.96	7	2516.57	0.10%
40	2	0.03	0.90	10	17.60	0.00%	0.92	10	86.90	0.00%	0.94	10	40.60	0.00%	0.96	7	3391.57	0.09%
40	2	0.05	0.90	10	13.70	0.00%	0.92	10	90.40	0.00%	0.94	10	20.40	0.00%	0.96	8	1026.62	0.09%
40	3	0.02	0.90	6	1258.33	0.27%	0.92	7	8256.43	1.29%	0.94	3	3788.00	1.06%	0.96			2.43%
40	3	0.03	0.90	6	1349.83	0.24%	0.92	6	3011.33	1.15%	0.94	3	5173.00	0.95%	0.96			2.43%
40	3	0.05	0.90	7	7175.29	0.29%	0.92	6	589.00	1.20%	0.94	3	11622.33	0.97%	0.96			2.55%
40	4	0.02	0.90	1	32151.00	2.61%	0.92			6.47%	0.94			4.38%	0.96			7.51%
40	4	0.03	0.90	1	19318.00	2.57%	0.92			6.06%	0.94			4.42%	0.96			7.02%
40	4	0.05	0.90			2.98%	0.92			6.89%	0.94			4.37%	0.96			7.28%
50	2	0.02	0.90	10	13.90	0.00%	0.92	9	2896.11	0.16%	0.94	10	123.00	0.00%	0.96	5	6185.40	0.32%
50	2	0.03	0.90	10	18.30	0.00%	0.92	9	3735.44	0.17%	0.94	10	112.10	0.00%	0.96	5	4966.80	0.33%
50	2	0.05	0.90	10	608.20	0.00%	0.92	8	110.00	0.19%	0.94	10	3282.60	0.00%	0.96	5	2819.20	0.40%
50	3	0.02	0.90	6	4342.17	1.09%	0.92	4	3233.50	0.88%	0.94	3	1926.67	1.02%	0.96			2.24%
50	3	0.03	0.90	6	4729.00	1.07%	0.92	4	3155.00	1.24%	0.94	3	1582.00	1.04%	0.96			2.16%
50	3	0.05	0.90	6	3547.17	1.10%	0.92	3	2296.00	1.07%	0.94	3	1080.00	1.03%	0.96			2.60%
50	4	0.02	0.90	2	2308.00	4.93%	0.92	1	12705.00	3.37%	0.94			2.83%	0.96			5.12%
50	4	0.03	0.90	2	1902.00	4.57%	0.92	1	12989.00	3.44%	0.94			3.16%	0.96			5.04%
50	4	0.05	0.90	2	7105.00	4.89%	0.92	1	15156.00	3.88%	0.94			3.31%	0.96			5.27%
60	2	0.02	0.90	10	1819.40	0.00%	0.92	8	26.38	0.05%	0.94	8	2549.50	0.10%	0.96	6	4313.17	0.12%
60	2	0.03	0.90	10	1558.70	0.00%	0.92	8	71.75	0.03%	0.94	9	4669.67	0.09%	0.96	6	6953.00	0.06%
60	2	0.05	0.90	10	1566.90	0.00%	0.92	8	50.50	0.07%	0.94	9	2291.00	0.09%	0.96	6	888.33	0.16%
60	3	0.02	0.90	3	2592.67	1.05%	0.92	2	6797.00	3.25%	0.94	3	18900.33	2.16%	0.96	1	196.00	2.77%
60	3	0.03	0.90	3	5301.33	1.18%	0.92	2	14896.50	3.14%	0.94	1	168.00	2.23%	0.96			3.40%
60	3	0.05	0.90	2	182.00	1.11%	0.92	1	429.00	3.04%	0.94	1	119.00	2.44%	0.96			3.38%
60	4	0.02	0.90	1	24866.00	2.66%	0.92			3.41%	0.94			4.40%	0.96			5.30%
60	4	0.03	0.90			2.69%	0.92			3.43%	0.94			4.33%	0.96			5.35%
60	4	0.05	0.90			3.14%	0.92			3.70%	0.94			4.14%	0.96			5.02%
Average					1578.80	2.56%			1560.94	3.86%			2097.01	3.24%			2698.15	5.30%
Total				204				175				168				108		

Table 3: Result of running the fixed policy π_2.

n	m	η	$\bar{f}$	Solved	Time(s)	Gap	$\bar{f}$	Solved	Time(s)	Gap	$\bar{f}$	Solved	Time(s)	Gap	$\bar{f}$	Solved	Time(s)	Gap
20	2	0.80	0.90	10	12.50	0.00%	0.92	10	16.00	0.00%	0.94	10	5.40	0.00%	0.96	10	263.20	0.00%
20	2	1.00	0.90	10	10.50	0.00%	0.92	10	12.50	0.00%	0.94	10	2.00	0.00%	0.96	10	13.40	0.00%
20	2	1.25	0.90	10	18.50	0.00%	0.92	10	28.50	0.00%	0.94	10	19.60	0.00%	0.96	10	59.50	0.00%
30	2	0.80	0.90	10	101.40	0.00%	0.92	9	3962.00	0.07%	0.94	10	1973.00	0.00%	0.96	8	646.75	0.22%
30	2	1.00	0.90	10	42.20	0.00%	0.92	9	3565.33	0.05%	0.94	10	890.70	0.00%	0.96	8	248.62	0.18%
30	2	1.25	0.90	10	193.60	0.00%	0.92	8	1.75	0.07%	0.94	9	3965.56	0.01%	0.96	8	1566.62	0.23%
40	2	0.80	0.90	10	32.70	0.00%	0.92	10	71.60	0.00%	0.94	10	26.80	0.00%	0.96	9	1671.78	0.04%
40	2	1.00	0.90	10	23.70	0.00%	0.92	10	34.60	0.00%	0.94	10	14.50	0.00%	0.96	10	1770.90	0.00%
40	2	1.25	0.90	10	41.50	0.00%	0.92	10	88.20	0.00%	0.94	10	55.90	0.00%	0.96	9	3170.89	0.07%
40	3	0.80	0.90	5	447.60	0.50%	0.92	7	3180.86	0.58%	0.94	5	9071.80	0.46%	0.96	1	10428.00	1.78%
40	3	1.00	0.90	6	4727.50	0.41%	0.92	7	908.00	0.49%	0.94	4	1945.25	0.86%	0.96	2	7175.00	1.39%
40	3	1.25	0.90	5	434.60	0.60%	0.92	5	4056.20	0.85%	0.94	3	6334.67	1.10%	0.96	1	1861.00	2.00%
40	4	0.80	0.90			2.47%	0.92			4.28%	0.94			3.81%	0.96			5.71%
40	4	1.00	0.90			2.37%	0.92			4.32%	0.94			3.02%	0.96			4.79%
40	4	1.25	0.90			3.08%	0.92			4.76%	0.94			4.74%	0.96			6.86%
50	2	0.80	0.90	10	113.50	0.00%	0.92	9	2252.00	0.00%	0.94	10	181.40	0.00%	0.96	7	7895.00	0.16%
50	2	1.00	0.90	10	124.40	0.00%	0.92	8	649.50	0.20%	0.94	10	86.40	0.00%	0.96	7	925.71	0.18%
50	2	1.25	0.90	10	84.40	0.00%	0.92	8	1980.50	0.20%	0.94	10	163.40	0.00%	0.96	7	2676.86	0.31%
50	3	0.80	0.90	4	1308.75	1.11%	0.92	4	4384.50	1.03%	0.94	3	1300.33	1.05%	0.96	1	2567.00	1.79%
50	3	1.00	0.90	5	3981.00	1.06%	0.92	5	5882.60	0.57%	0.94	4	8905.00	0.77%	0.96	1	127.00	1.39%
50	3	1.25	0.90	4	3068.00	1.16%	0.92	5	8601.00	0.86%	0.94	4	8774.00	1.13%	0.96	1	349.00	1.84%
50	4	0.80	0.90	2	124.00	4.31%	0.92	2	11846.50	2.77%	0.94			2.27%	0.96			3.89%
50	4	1.00	0.90	2	85.50	3.99%	0.92	2	7662.00	2.89%	0.94			2.01%	0.96			3.53%
50	4	1.25	0.90	2	164.00	4.97%	0.92	2	17078.50	3.33%	0.94			2.76%	0.96			4.46%
60	2	0.80	0.90	10	1561.70	0.00%	0.92	9	1438.22	0.06%	0.94	7	486.71	0.09%	0.96	7	5270.29	0.14%
60	2	1.00	0.90	10	1035.50	0.00%	0.92	9	1047.22	0.02%	0.94	8	3190.88	0.06%	0.96	8	3104.75	0.13%
60	2	1.25	0.90	10	1813.00	0.00%	0.92	9	961.89	0.06%	0.94	8	4489.38	0.10%	0.96	7	3242.14	0.21%
60	3	0.80	0.90	4	5910.00	0.89%	0.92	1	407.00	2.55%	0.94	2	12122.50	2.01%	0.96	1	2326.00	2.59%
60	3	1.00	0.90	3	2109.00	0.89%	0.92	2	597.00	2.29%	0.94	2	4097.00	1.90%	0.96	2	4511.50	2.22%
60	3	1.25	0.90	4	3508.25	0.92%	0.92	1	224.00	2.97%	0.94	1	967.00	2.21%	0.96	1	581.00	2.66%
60	4	0.80	0.90			2.30%	0.92			2.90%	0.94			3.75%	0.96			4.06%
60	4	1.00	0.90			2.21%	0.92			2.41%	0.94			3.38%	0.96			3.75%
60	4	1.25	0.90			2.57%	0.92			2.92%	0.94			4.43%	0.96			4.70%
Average					852.17	2.39%			1969.43	2.90%			1852.34	2.80%			2138.75	4.08%
Total				196				181				170				136		

Table 4: Result of running the fixed policy π_3.

n	m	λ	$\bar{f}$	Solved	Time(s)	Gap	$\bar{f}$	Solved	Time(s)	Gap	$\bar{f}$	Solved	Time(s)	Gap	$\bar{f}$	Solved	Time(s)	Gap
20	2	0.80	0.90	10	14.80	0.00%	0.92	10	18.40	0.00%	0.94	10	8.30	0.00%	0.96	9	1147.00	0.17%
20	2	0.90	0.90	10	31.30	0.00%	0.92	10	25.50	0.00%	0.94	10	74.90	0.00%	0.96	7	330.14	0.63%
20	2	1.00	0.90	10	195.40	0.00%	0.92	10	35.60	0.00%	0.94	9	260.11	0.00%	0.96	5	8180.20	1.24%
30	2	0.80	0.90	10	19.40	0.00%	0.92	9	508.56	0.04%	0.94	10	1235.90	0.00%	0.96	8	3460.38	0.27%
30	2	0.90	0.90	10	24.00	0.00%	0.92	9	482.00	0.15%	0.94	10	3835.10	0.00%	0.96	5	1732.80	0.79%
30	2	1.00	0.90	10	228.90	0.00%	0.92	9	529.11	0.16%	0.94	9	5148.00	0.05%	0.96	3	1992.67	1.73%
40	2	0.80	0.90	10	11.70	0.00%	0.92	10	202.30	0.00%	0.94	10	66.90	0.00%	0.96	9	4205.78	0.04%
40	2	0.90	0.90	10	26.90	0.00%	0.92	10	269.90	0.00%	0.94	10	106.30	0.00%	0.96	5	11483.60	0.23%
40	2	1.00	0.90	10	584.30	0.00%	0.92	10	703.40	0.00%	0.94	9	727.78	0.00%	0.96	1	2025.00	1.20%
40	3	0.80	0.90	5	2571.60	0.22%	0.92	6	3434.67	1.35%	0.94	2	8629.50	1.75%	0.96			2.96%
40	3	0.90	0.90	6	2744.50	0.31%	0.92	6	3604.50	1.31%	0.94	2	16319.00	2.83%	0.96			4.17%
40	3	1.00	0.90	6	4285.17	0.42%	0.92	6	6498.17	1.69%	0.94	1	5093.00	3.81%	0.96			7.14%
40	4	0.80	0.90	1	14852.00	3.91%	0.92			7.34%	0.94			4.70%	0.96			7.86%
40	4	0.90	0.90			5.09%	0.92			8.99%	0.94			5.67%	0.96			9.64%
40	4	1.00	0.90			6.46%	0.92			10.32%	0.94			7.60%	0.96			11.78%
50	2	0.80	0.90	10	1515.90	0.00%	0.92	8	1761.12	0.17%	0.94	10	312.10	0.00%	0.96	5	4882.80	0.31%
50	2	0.90	0.90	10	935.20	0.00%	0.92	7	135.71	0.19%	0.94	9	554.11	0.04%	0.96	2	781.00	0.53%
50	2	1.00	0.90	10	2957.20	0.00%	0.92	8	3506.00	0.17%	0.94	9	7396.44	0.06%	0.96			1.40%
50	3	0.80	0.90	6	3488.67	1.10%	0.92	4	4239.50	1.38%	0.94	3	2491.33	0.87%	0.96			3.45%
50	3	0.90	0.90	5	6659.20	1.17%	0.92	3	2304.67	1.46%	0.94	3	6620.33	1.43%	0.96			4.66%
50	3	1.00	0.90	6	8170.00	1.18%	0.92	3	9284.33	1.67%	0.94	1	26215.00	1.67%	0.96			7.46%
50	4	0.80	0.90	2	4060.00	4.74%	0.92	1	10983.00	5.94%	0.94			4.28%	0.96			5.59%
50	4	0.90	0.90	2	2043.50	5.41%	0.92	1	16596.00	7.68%	0.94			5.22%	0.96			7.23%
50	4	1.00	0.90	2	1767.00	6.29%	0.92	1	23014.00	8.74%	0.94			7.12%	0.96			10.88%
60	2	0.80	0.90	10	1667.60	0.00%	0.92	9	2333.11	0.05%	0.94	9	3909.44	0.07%	0.96	5	4470.40	0.19%
60	2	0.90	0.90	10	1579.70	0.00%	0.92	8	358.25	0.05%	0.94	9	7823.11	0.10%	0.96	3	6583.33	0.42%
60	2	1.00	0.90	10	2323.80	0.00%	0.92	8	627.25	0.06%	0.94	6	160.83	0.11%	0.96			1.21%
60	3	0.80	0.90	4	4117.50	1.02%	0.92	1	1145.00	3.01%	0.94	2	18532.50	2.60%	0.96			3.50%
60	3	0.90	0.90	4	10966.50	1.22%	0.92	2	17711.00	3.51%	0.94	1	3175.00	3.06%	0.96			4.28%
60	3	1.00	0.90	2	8169.50	1.81%	0.92	1	14206.00	4.84%	0.94	1	14036.00	4.04%	0.96			6.08%
60	4	0.80	0.90			2.68%	0.92			3.49%	0.94			4.78%	0.96			6.01%
60	4	0.90	0.90			3.38%	0.92			4.34%	0.94			6.04%	0.96			7.61%
60	4	1.00	0.90			4.20%	0.92			5.68%	0.94			8.22%	0.96			9.56%
Average					1923.95	3.37%			1955.95	5.59%			2919.66	5.07%			3899.00	8.68%
Total				201				170				155				67		

when solving the VRPSD under policy π_2. Furthermore, with the exception of the instances where $\bar{f} = 0.92$, the use of policy π_2 also enables the smallest weighted average running times to be obtained when applying the algorithm i.e., 852.17 seconds, 1,852.34 seconds and 2,138.75 seconds for the instances where $\bar{f} = 0.90$, $\bar{f} = 0.94$ and $\bar{f} = 0.96$, respectively. In the case of the instances where $\bar{f} = 0.92$, policy π_1 allows the Integer L-shaped algorithm to be more computationally efficient (i.e., a weighted average of 1 560.94 seconds was obtained using π_1, compared to 1 969.43 seconds using π_2). However, when comparing policies using the computation times obtained by the algorithm, it is important to note that the reported results are not perfectly comparable considering that they are not necessarily based on runs performed on the same instances. For example, the weighted average obtained for policy π_1 on the instances where $\bar{f} = 0.92$ is based on less instances solved to optimality when compared to π_2 (i.e., 175 instances in the case of π_1 versus 181 instances in the case of π_2). This being said, what these results show is again the trend that the Integer L-shaped algorithm is most efficient under policy π_2 to solve the VRPSD.

Finally, when considering the average gaps obtained when applying the different policies, the use of π_2 provides again the best results. For the different filling coefficient values defining the considered instances (i.e., $\bar{f}$ = 0.90, 0.92, 0.94 and 0.96), the average gaps obtained overall runs are respectively: 2.56%, 3.86%, 3.24% and 5.30% when applying π_1; 2.39%, 2.90%, 2.80% and 4.08% when applying π_2; and 3.37%, 5.59%, 5.07% and 8.68% when applying π_3. Therefore, one can conclude that the overall numerical complexity of solving the VRPSD using the Integer L-shaped algorithm seems easiest using π_2, followed by π_1 and π_3. In addition, policy π_3 appears as the most challenging to apply when considering all previously analyzed metrics.

5.2 Solution Cost Assessment

In this subsection, we analyze how the three proposed policies perform in terms of reducing the costs associated with the vehicle routes. Given that a company may choose to use any of the policies based on the specific operational rules that are applied to perform the routes, it is important to note that our aim here is not necessarily to identify which policy is best overall. Instead, we will analyze the quality of the solutions obtained using π_1, π_2 and π_3 by evaluating them under both the classical recourse and the optimal restocking policies. By doing so, for the solutions obtained, we will assess how π_1, π_2 and π_3 1) reduce the number of failures when compared to applying the routes using the classical recourse policy and 2) approximate the optimal restocking cost.

Therefore, when solving the instances using the three proposed policies, we first consider only those runs where optimal solutions were found. The routes associated with these optimal solutions are then alternatively evaluated using both the classical recourse and optimal restocking policies, the latter was computed similar to Bertsimas et al. (1995). Also, results will be grouped according to the filling rate $\bar{f}$ of the instances, which is a problem dimension that clearly impacts the numerical challenges involved in solving the instances. In Table 5, we first report the ratios obtained between the expected number of BF trips that are performed when the routes are conducted under the classical recourse policy (i.e., EBF_c) with respect to when they are performed under the proposed rule-based policies (i.e., EBF_r).

As shown in Table 5, compared to the classical recourse policy, the use of π_1, π_2 and π_3 clearly reduces the expected number of BF trips that are performed when applying the routes. Given the practical high costs that may be associated with the disturbances related to route failures, the proposed policies offer a clear advantage over the myopic classical recourse policy. In addition, when analyzing the results obtained for π_1 and π_2, one sees how the use of more risk-averse preset values can further reduce the expected number of performed BF trips. A significant reduction is observed when π_2 is applied using $\eta = 1.25$ in which case the average ratios increase by an order of magnitude. Regarding policy π_3, the obtained results seem to contradict these observations. However, this can be explained by the fact that, for a given instance type (i.e., for fixed parameters n, m and $\bar{f}$), the value to which λ is fixed greatly influences the number of instances solved to optimality. From Table 4, one observes the trend that the VRPSD becomes significantly harder to solve as the value λ is increased when applying π_3. Therefore, in this case, the average ratios are computed using the solutions obtained on noticeably different sets of instances which, in turn, can explain the differing observations.

Table 5: The ratio $\frac{EBF_c}{EBF_r}$

π	*preset*	$\bar{f} = 0.90$	$\bar{f} = 0.92$	$\bar{f} = 0.94$	$\bar{f} = 0.96$
	$\delta = 0.02$	1.63	1.54	1.83	1.44
π_1	$\delta = 0.03$	3.75	1.50	1.97	2.03
	$\delta = 0.05$	5.04	2.33	3.48	3.28
	$\eta = 0.80$	2.31	2.27	2.16	2.25
π_2	$\eta = 1.00$	4.19	4.93	4.53	5.37
	$\eta = 1.25$	35.55	27.95	40.96	44.76
	$\lambda = 0.80$	25.97	6.89	10.12	3.56
π_3	$\lambda = 0.90$	13.95	2.71	7.36	2.58
	$\lambda = 1.00$	13.75	6.16	6.52	1.35

The final step in our overall analysis is to assess how policies π_1, π_2 and π_3 impact the solution costs. In Table 6, for those instances solved to optimality, the average relative differences are reported between the solution costs obtained by using the rule-based policies and both the classical recourse (i.e., the Savings columns) and the optimal restocking policies (i.e., the Deviations columns). Therefore, the Savings values indicate the relative reductions in terms of solution cost that are obtained when the routes are applied using the proposed rule-based policies, when compared to the classical recourse policy. As for the Deviations values, they represent the gap between the solution cost evaluated using the rule-based policies and the optimal restocking policy on the same routes. It should be noted that, for a given route, the optimal restocking cost defines a lower bound over all possible policies.

When analyzing these results, one first notices that the values obtained are relatively small. This can be explained by the fact that the policies are being evaluated on the same routes coupled with the fact that the value b is not severely penalizing route failures. This being said, with the exception of π_3 on three distinct instance categories (i.e., when solving the $\bar{f} = 0.90$ instances with $\lambda = 1.00$ and the $\bar{f} = 0.96$ instances with $\lambda = 0.90$ and $\lambda = 1.00$), all ruled-based policies when applied on the obtained routes provide a cost reduction (or are equivalent) when compared to the classical recourse policy. The best savings are obtained for π_2 on the $\bar{f} = 0.96$ instances. Furthermore, the observed savings tend to increase as the value of $\bar{f}$ increases also. This is to be expected given the positive correlation that exists between the expected number of failures and the overall filling coefficient of instances. Regarding policy π_3, the three observed exceptions may be explained by an overly risk-averse implementation of the policy which occurs by fixing the preset value to $\lambda = 0.90$ and $\lambda = 1.00$. Considering that these runs produce savings that are extremely small when compared to other policy runs, one can infer that the number of PR trips that are performed in an effort to reduce the number of failures, in these cases, does not seem to provide an added overall cost advantage.

Table 6: Savings and Deviations.

		$\bar{f} = 0.90$		$\bar{f} = 0.92$		$\bar{f} = 0.94$		$\bar{f} = 0.96$	
π	*preset*	Savings	Deviations	Savings	Deviations	Savings	Deviations	Savings	Deviations
π_1	$\delta = 0.02$	0.04%	0.07%	0.06%	0.13%	0.11%	0.34%	0.35%	0.86%
	$\delta = 0.03$	0.08%	0.05%	0.04%	0.11%	0.13%	0.34%	0.46%	0.73%
	$\delta = 0.05$	0.00%	0.12%	0.05%	0.13%	0.16%	0.38%	0.43%	0.81%
π_2	$\eta = 0.80$	0.11%	0.04%	0.19%	0.09%	0.35%	0.18%	0.81%	0.45%
	$\eta = 1.00$	0.18%	0.01%	0.39%	0.01%	0.61%	0.03%	1.29%	0.08%
	$\eta = 1.25$	0.07%	0.05%	0.09%	0.14%	0.47%	0.25%	1.05%	0.38%
π_3	$\lambda = 0.80$	0.05%	0.06%	0.08%	0.42%	0.22%	0.25%	0.50%	0.68%
	$\lambda = 0.90$	0.00%	0.09%	0.00%	0.46%	0.02%	0.34%	−0.02%	1.27%
	$\lambda = 1.00$	−0.04%	0.10%	0.06%	0.48%	0.27%	0.60%	−1.57%	3.14%

Finally, when comparing the proposed policies to the optimal restocking one, it can be observed that the relative differences are quite small. Policy π_2 appears as the best to approximate the optimal restocking cost for the considered solutions. Specifically, when the policy is applied with its preset value fixed to $\eta = 1.00$, the average deviations vary between 0.01% and 0.08%. Therefore, such a policy provides a very good approximation for the optimal restocking cost. Furthermore, when compared to both π_1 and π_3, when π_2 is applied on instances for increasing values of $\bar{f}$, one observes an increase in the deviation values (i.e., a deterioration of the approximation) but at much less pronounced rate. Comparatively, π_3 appears as the worst policy to approximate the optimal restocking cost. However, this can again be explained by the overly risk-averse implementations of the policy.

6 Conclusions

In this paper, we introduce a new type of recourse policies for the VRPSD, that are based on the use of a set of fixed operational rules, specifying when both PR and BF trips need to be performed. Given a route, such policies can be expressed as a set of thresholds, associated with each customer scheduled along the route, that define when PR trips need to be performed. We also show how the recourse cost of routes can be efficiently computed using a recursive function based on the obtained thresholds. Finally, we propose an exact solution method, using the Integer L-shaped algorithm, to solve the considered problem. With our solution method, problems with up to 60 customers and a fleet of four vehicles are solved to optimality.

Through our extensive numerical experiments, we show that the defined ruled-based policies outperform the classical policy in terms of reducing the number of failures occurring when implementing routes and their associated costs. Furthermore, it is also observed that the overall cost of the routes, when computed using an optimal restocking policy, remain close to the cost originally obtained using the ruled-based policies. Clearly demonstrating that the proposed policies also define a good approximation to the optimal one. Finally, the proposed solution method is numerically shown to be efficient to tackle a wide range of problems of varying size and for different filling rates.

The present paper has defined a series of interesting avenues of research. Namely, other families of rule-based policies can be defined. These should capture other operational rules likely to be used in practice.

Acknowledgment

This research was partially funded by the Natural Sciences and Engineering Research Council of Canada under the discovery grants program. This support is gratefully acknowledged.

References

Bertsimas, Dimitris, Philippe Chervi, Michael Peterson. 1995. Computational approaches to stochastic vehicle routing problems. *Transportation science* **29**(4) 342–352.

Bertsimas, Dimitris J, Patrick Jaillet, Amedeo R Odoni. 1990. A priori optimization. *Operations Research* **38**(6) 1019–1033.

Bianchi, Leonora. 2006. Ant colony optimization and local search for the probabilistic traveling salesman problem: a case study in stochastic combinatorial optimization. Ph.D. thesis, Universite Libre de Bruxelles, Brussels, Belgium.

Christiansen, Christian H, Jens Lysgaard. 2007. A branch-and-price algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research Letters* **35**(6) 773–781.

Dantzig, George B, John H Ramser. 1959. The truck dispatching problem. *Management science* **6**(1) 80–91.

Dror, Moshe, Gilbert Laporte, Pierre Trudeau. 1989. Vehicle routing with stochastic demands: Properties and solution frameworks. *Transportation science* **23**(3) 166–176.

Dror, Moshe, Pierre Trudeau. 1986. Stochastic vehicle routing with modified savings algorithm. *European Journal of Operational Research* **23**(2) 228–235.

Gauvin, Charles, Guy Desaulniers, Michel Gendreau. 2014. A branch-cut-and-price algorithm for the vehicle routing problem with stochastic demands. *Computers & Operations Research* **50** 141–153.

Gendreau, Michel, Ola Jabali, Walter Rei. 2014. Stochastic vehicle routing problems. Paolo Toth, Daniele Vigo, eds., *Vehicle routing: Problems, methods, and applications*, second edition ed., chap. 8. MOS-SIAM Series on Optimization, SIAM, Philadelphia, 213–239.

Gendreau, Michel, Ola Jabali, Walter Rei. 2016. 50th anniversary invited articlefuture research directions in stochastic vehicle routing. *Transportation Science* **50**(4) 1163–1173.

Gendreau, Michel, Gilbert Laporte, René Séguin. 1995. An exact algorithm for the vehicle routing problem with stochastic demands and customers. *Transportation science* **29**(2) 143–155.

Gendreau, Michel, Gilbert Laporte, René Séguin. 1996. A tabu search heuristic for the vehicle routing problem with stochastic demands and customers. *Operations Research* **44**(3) 469–477.

Gendron, Bernard, Teodor Gabriel Crainic, Antonio Frangioni, François Guertin. 2005. Oobb: Object-oriented tools for parallel branch-and-bound. *PAREO*.

Hjorring, Curt, John Holt. 1999. New optimality cuts for a single-vehicle stochastic routing problem. *Annals of Operations Research* **86** 569–584.

Jabali, Ola, Walter Rei, Michel Gendreau, Gilbert Laporte. 2014. Partial-route inequalities for the multi-vehicle routing problem with stochastic demands. *Discrete Applied Mathematics* **177** 121–136.

Kovacs, Attila A, Bruce L Golden, Richard F Hartl, Sophie N Parragh. 2014. Vehicle routing problems in which consistency considerations are important: A survey. *Networks* **64**(3) 192–213.

Laporte, Gilbert, François V Louveaux. 1993. The integer l-shaped method for stochastic integer programs with complete recourse. *Operations research letters* **13**(3) 133–142.

Laporte, Gilbert, François V Louveaux, Luc Van Hamme. 2002. An integer l-shaped algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research* **50**(3) 415–423.

Louveaux, François V. 1998. An introduction to stochastic transportation models. Martine Labbe, Gilbert Laporte, Katalin Tanczos, Philippe Toint, eds., *Operations Research and Decision Aid Methodologies in Traffic and Transportation Management*, *NATO ASI Series*, vol. 166. Springer Berlin Heidelberg, 244–263.

Lysgaard, Jens, Adam N Letchford, Richard W Eglese. 2004. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming* **100**(2) 423–445.

Mendoza, Jorge E, Juan G Villegas. 2013. A multi-space sampling heuristic for the vehicle routing problem with stochastic demands. *Optimization Letters* **7**(7) 1503–1516.

Novoa, Clara, Robert Storer. 2009. An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. *European Journal of Operational Research* **196**(2) 509–515.

Rei, Walter, Michel Gendreau, Patrick Soriano. 2010. A hybrid monte carlo local branching algorithm for the single vehicle routing problem with stochastic demands. *Transportation Science* **44**(1) 136–146.

Secomandi, Nicola. 2000. Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands. *Computers & Operations Research* **27**(11) 1201–1225.

Secomandi, Nicola. 2001. A rollout policy for the vehicle routing problem with stochastic demands. *Operations Research* **49**(5) 796–802.

Secomandi, Nicola, Francois Margot. 2009. Reoptimization approaches for the vehicle-routing problem with stochastic demands. *Operations Research* **57**(1) 214–230.

Toth, Paolo, Daniele Vigo. 2014. *Vehicle Routing: Problems, Methods, and Applications*, vol. 18. SIAM.

Yang, Wen-Huei, Kamlesh Mathur, Ronald H Ballou. 2000. Stochastic vehicle routing problem with restocking. *Transportation Science* **34**(1) 99–112.

Yee, James R, Bruce L Golden. 1980. A note on determining operating strategies for probabilistic vehicle routing. *Naval Research Logistics Quarterly* **27**(1) 159–163.