Université de Montréal

Recourse Policies in Vehicle Routing Problem with Stochastic Demands

par
Majid Salavati-Khoshghalb

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Thèse présentée à la Faculté des études supérieures
en vue de l'obtention du grade de Philosophiæ Doctor (Ph.D.)
en informatique option recherche opérationnelle

August, 2017

Université de Montréal
Faculté des études supérieures

Cette thèse intitulée:

Recourse Policies in Vehicle Routing Problem with Stochastic Demands

présentée par:

Majid Salavati-Khoshghalb

a été évaluée par un jury composé des personnes suivantes:

Bernard Gendron,	président-rapporteur
Michel Gendreau,	directeur de recherche
Walter Rei,	codirecteur
Ola Jabali,	codirecteur
Pierre L'Écuyer,	membre du jury
Fernando Ordóñez,	examinateur externe
Bernard Gendron,	représentant du doyen de la FES

Thèse acceptée le: 29/August/2017

RÉSUMÉ

Dans le domaine de la logistique, de nombreux problèmes pratiques peuvent être formulés comme le problème de tournées de véhicules (PTV). Dans son image la plus large, le PTV vise à concevoir un ensemble d'itinéraires de collecte ou de livraison des marchandises à travers un ensemble de clients avec des coûts minimaux. Dans le PTV déterministe, tous les paramètres du problème sont supposés connus au préalable. Dans de nombreuses variantes de la vie réelle du PTV, cependant, ils impliquent diverses sources d'aléatoire. Le PTV traite du caractère aléatoire inhérent aux demandes, présence des clients, temps de parcours ou temps de service. Les PTV, dans lesquels un ou plusieurs paramètres sont stochastiques, sont appelés des problèmes stochastiques de tournées de véhicules (PSTV).

Dans cette dissertation, nous étudions spécifiquement le problème de tournées de véhicules avec les demandes stochastiques (PTVDS). Dans cette variante de PSTV, les demandes des clients ne sont connues qu'en arrivant à l'emplacement du client et sont définies par des distributions de probabilité. Dans ce contexte, le véhicule qui exécute une route planifiée peut ne pas répondre à un client, lorsque la demande observée dépasse la capacité résiduelle du véhicule. Ces événements sont appelés les échecs de l'itinéraire ; dans ce cas, l'itinéraire planifié devient non-réalisable. Il existe deux approches face aux échecs de l'itinéraire. Au client où l'échec s'est produit, on peut récupérer la realisabilite en exécutant un aller-retour vers le dépôt, pour remplir la capacité du véhicule et compléter le service. En prévision des échecs d'itinéraire, on peut exécuter des retours préventifs lorsque la capacité résiduelle est inférieure à une valeur seuil. Toutes les décisions supplémentaires, qui sont sous la forme de retours au dépôt dans le contexte PTVDS, sont appelées des actions de recours. Pour modéliser le PTVDS, une politique de recours, régissant l'exécution des actions de recours, doit être conçue.

L'objectif de cette dissertation est d'élaborer des politiques de recours rentables, dans lesquelles les conventions opérationnelles fixes peuvent régir l'exécution des actions de recours. Nous fournissons un cadre général pour classer les conventions opérationnelles fixes pour être utilisées dans le cadre PTVDS. Dans cette classification, les conventions opérationnelles fixes peuvent être regroupées dans (i) les politiques basées sur le volume, (ii) les politiques basées sur le risque et (iii) les politiques basées sur le distance. Les politiques hybrides, dans lesquelles plusieurs règles fixes sont incorporées, peuvent être envisagées. Dans la première partie de cette thèse, nous proposons une politique fixe basée sur les règles, par laquelle l'exécution des retours préventifs est régie par les seuils prédéfinis. Nous proposons notamment trois politiques basées sur le volume qui tiennent compte de la capacité du véhicule, de la demande attendue du prochain client et de la demande attendue des clients non visités. La méthode “Integer L-Shaped" est réaménagée pour résoudre le PTVDS selon la politique basée sur les règles.

Dans la deuxième partie, nous proposons une politique de recours hybride, qui combine le risque d'échec et de distance à parcourir en une seule règle de recours, régissant l'exécution des recours. Nous proposons d'abord une mesure de risque pour contrôler le risque d'échec au prochain client. Lorsque le risque d'échec n'est ni trop élevé ni trop bas, nous utilisons une mesure de distance, ce qui compare le coût de retour préventif avec les coûts d'échecs futurs.

Dans la dernière partie de cette thèse, nous développons une méthodologie de solution exacte pour résoudre le VRPSD dans le cadre d'une politique de restockage optimale. La politique de restockage optimale résulte d'un ensemble de seuils spécifiques au client, de sorte que le coût de recours prévu soit réduit au minimum.

Mots clés : Problème de tournées de véhicules avec les demandes stochastiques, recours, politique basée sur les règles, hybride, politique de restockage optimale, méthodologie de solution exacte.

ABSTRACT

In the field of logistics, many practical problems can be formulated as the vehicle routing problem (VRP). In its broadest picture, the VRP aims at designing a set of vehicle routes to pickup or delivery goods through a set of customers with the minimum costs. In the deterministic VRP, all problem parameters are assumed known beforehand. The VRPs in real-life applications, however, involve various sources of uncertainty. Uncertainty is appeared in several parameters of the VRPs like demands, customer, service or traveling times. The VRPs in which one or more parameters appear to be uncertain are called stochastic VRPs (SVRPs).

In this dissertation, we examine vehicle routing problem with stochastic demands (VRPSD). In this variant of SVRPs, the customer demands are only known upon arriving at the customer location and are defined through probability distributions. In this setting, the vehicle executing a planned route may fail to service a customer, whenever the observed demand exceeds the residual capacity of the vehicle. Such occurrences are called route failures; in this case the planned route becomes infeasible. There are two approaches when facing route failures. At the customer where the failure occurred, one can recover routing feasibility by executing back-and-forth trips to the depot to replenish the vehicle capacity and complete the service. In anticipation of route failures, one can perform preventive returns whenever the residual capacity falls below a threshold value. All the extra decisions, which are in the form of return trips to the depot in the VRPSD context, preserving routing feasibility are called recourse actions. To model the VRPSD, a recourse policy, governing the execution of such recourse actions, must be designed. The goal of this dissertation is to develop cost-effective recourse policies, in which the fixed operational conventions can govern the execution of recourse actions.

In the first part of this dissertation, we propose a fixed rule-based policy, by which the execution of preventive returns is governed through the preset thresholds. We particularly introduce three volume based policies which consider the vehicle capacity, expected demand of the next customer and the expected demand of the remaining unvisited customers. Then, the Integer L-shaped algorithm is redeveloped to solve the VRPSD under the rule-based policy. The contribution with regard to this study has been submitted to the Journal of Transportation Science.

In the second part, we propose a hybrid recourse policy, which combines the risk of failure and distances-to-travel into a single recourse rule, governing the execution of recourse actions. We employ a risk measure to control the risk of failure at the next customer. When the risk of failure is neither too high nor too low, we apply a distance measure, which compares the preventive return cost with future failures cost. The contribution with regard to this study has been submitted to the EURO Journal on Transportation and Logistics.

In the last part of this dissertation, we develop an exact solution methodology to solve the VRPSD under an optimal restocking policy. The optimal restocking policy derives a set of customer-specific thresholds such that the expected recourse cost is minimized. The contribution with regard to this study will be submitted to the European Journal of Operational Research.

Keywords: Vehicle routing problem with stochastic demands, recourse, rule-based policy, hybrid, optimal restocking policy, exact method.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

VRP	Vehicle Routing Problem
TSP	Traveling Salesman Problem
SVRP	Stochastic Vehicle Routing Problem
VRPSD	Vehicle Routing Problem with Stochastic Demands
BF	Back-and-Forth
PR	Preventive Restocking
SPR	Stochastic Programming with Recourse
TS-SIPR	Two-Stage Stochastic Integer Program with Recourse
VRPSDC	Vehicle Routing Problem with Stochastic Demands and Customers
CCP	Chance-Constrained Programming
CP	Current Problem
B&C	Branch-and-Cut
LBF	Lower Bounding Functional
MDP	Markov Decision Process
SSPP	Stochastic Shortest Path Problem
RVRP	Robust Vehicle Routing Problem

To my parents and my lovely wife.

ACKNOWLEDGMENTS

I would like to take this opportunity to express my gratitude to everyone who helped me throughout the course of my Ph.D.

First of all, I cannot express enough my gratitude to my supervisor, Professor Michel Gendreau, for his continuous inspirations, expertise and support. It was a real honor to work under your supervision. Also, I would like to express my deepest appreciation to my co-supervisors, Professor Walter Rei and Professor Ola Jabali, for their invaluable guidance, encouragements and patience.

I am also so grateful to the members of my committee for their detailed and insightful comments on the draft of this dissertation.

A very special gratitude goes out to all my friends and staffs at UdeM and CIRRELT, specifically Serge Bisaillon, who constantly helped me through the implementation of my algorithms.

I would also greatly appreciate the funding support provided by NSERC and UdeM throughout my doctoral studies.

Last but not least, none of this could have happened without my family. I want to express my deepest gratitude to my parents for all of the sacrifices that you've made on my behalf. And most of all, I would like to thank my beloved wife Fatemeh for her love, outstanding support and lovely smile, which are the greatest gift I was given in this journey.

CHAPTER 1

INTRODUCTION

Combinatorial optimization is one of the most important fields that is shared by operations research and computer science disciplines. One of the classic combinatorial optimization problems that has received considerable interest since it's been introduced by Dantzig and Ramser [15], is the *Vehicle Routing Problem* (VRP). The Mathematical model discussed by Dantzig and Ramser [15] is a generalization of the *Traveling Salesman Problem* (TSP), see Flood [18]. It should be noted that the VRP is an NP-hard problem, because it generalizes the TSP, e.g., see Gutin and Punnen [26] for the complexity of TSP.

The VRP aims at planning a set of routes with the minimum travel cost for a fleet of identical vehicles, dispatching from a depot location to visit a set of customers (each customer exactly once) with deterministic demands, in order to pickup or delivery. Technological advances in the power of current computer systems enable researchers to solve more complex problems in term of size of VRP instances. In addition, the significant advances in the modeling of problems and the need to explicitly consider more realistic aspects of real-life problems, like time constraints, multi compartment services, simultaneous pickup and delivery, etc., encourage researchers to develop sophisticated solution methods to treat recent variants of VRP, see e.g., Toth and Vigo [55] for a thorough exposition in the progress of modeling frameworks and solution approaches.

In the *Deterministic VRP*, all parameters associated to the activities, services, and resources are fully known beforehand. In real life problems, however, the VRP is subject of various sources of uncertainty which may be appeared in demands, travel and service times. The VRPs in which one deals with some sources of stochasticity are called *Stochastic VRPs* (SVRPs). In the stochastic environment, one may use the deterministic approximated models in which the existing source of uncertainty is replaced by a roughly estimation or forecast; for instance one can replaces the stochastic demand of a customer by the expected demand. These approximated models generally produce bad quality solutions which become inefficient in the execution time; unexpected extra costs will be incurred by stochastic events, e.g., see Louveaux [36] for the discussion of advantages of using the stochastic models. Therefore, there is an apparent need to propose and develop: (i) stochastic optimization models that explicitly consider stochasticity using random variables, and (ii) solution methodologies to solve such stochastic optimization models, efficiently.

This dissertation deals with a variant of SVRPs, called the *VRP with Stochastic Demands* (VRPSD), in which the customer demands are stochastic and are only known as probability distributions, beforehand. In this research, the stochastic demands follow general discrete distributions. In the context of VRPSD, a planned route may fail and become infeasible, i.e., when the vehicle executing the route visits a customer with an excessive demand, i.e., a route *failure* is occurred. In a failure event the service is interrupted and route becomes infeasible. There are two ways to deal with demand uncertainty in the VRPSD. One can construct feasible routing decisions such that the maximum demand of customers can be serviced by the fleet of vehicles. However, in practice, this approach may result in inefficient routes. As an alternative approach to tackle stochasticity, one can construct routes in such a way that the vehicles, executing the route and confronting various demand observations, are able to make corrective decisions, which preserve route feasibility, when facing route failures. In this approach, the vehicle may perform corrective extra return trips as the *recourse actions* which entail additional costs called *recourse costs*. Depending on how routing and recourse decisions are made, and how stochastic demands are observed, there are two main paradigm to model the VRPSD, see Gendreau et al. [22]. In the first approach, called the *a priori optimization* approach, one can partition the overall decision making process into two stages; in the first stage one can construct the routing decisions; and when demand uncertainty reveals itself in the execution of routing decisions in the second stage, the vehicle preserves route feasibility whenever it is needed by executing return trips to the depot. As an alternative, in the *re-optimization* approach one generally makes routing and recourse decisions (which customer is potentially the next customer to proceed and what recourse action must be taken before visiting the next customer) at each customer such that the optimal cost-to-go is minimized. In the rest of this dissertation, we model the VRPSD under the a priori optimization approach by which a first-stage routing decisions is obtained beforehand.

There are two approaches to deal with route failures. One can recover routing feasibility by executing a *Back-Forth* (BF) trip to the depot to replenish the capacity, as a *reactive* recourse action, once a failure event is occurred at a specific customer. After replenishing the capacity at the depot, the vehicle completes the service at the customer at where the failure occurred, and continues the service for the remaining unvisited customers. In anticipation of potential failures at the next customer or following customers, one can alternatively prescribe a *Preventive Restocking* (PR) trip in a *proactive* fashion. In such a case, instead of visiting the next customer, the vehicle with a low-stock capacity replenishes the vehicle capacity at the depot and then visits the next customer. It should be noted that if the residual capacity of the vehicle and the observed demand are equal, an *exact stockout* is occurred. In such a case, the vehicle after serving the current customer may perform *restocking trip* to replenish the vehicle capacity.

To formulate the VRPSD, a *recourse policy*, which governs how recourse decisions (by means of a set of predetermined recourse actions) are taken when facing stochastic events, must be determined. In this dissertation we model the VRPSD using the *a priori optimization* modeling paradigm proposed by Dror et al. [17] and Bertsimas et al. [4]. In such setting, a set of vehicle routes must be planned to be executed in a long planning horizon. Depending on how the various decisions (both routing and recourse) are made in the VRPSD (i.e., either statically or dynamically), the solution approaches proposed for the problem can be classified as in Table 1.I.

Considering the stochastic nature of the problem, the vehicle may fail to service customers repeatedly. From the customer's perspective, two consecutive visits to fulfill demands causes disturbance. Therefore, route failures can significantly interrupt the service resulting in unsatisfactory, lengthen the planned routes, and cause arrival time consistency issues. The aim of this dissertation is to propose various new efficient recourse policies which incorporate PR trips in order to reduce the risk of observing route failures.

In the following, we briefly discuss existing recourse policies, their advantages and related drawbacks. The most studied recourse policy in the context of the VRPSD is the *Classical Recourse* which consists of following the planned route until a route failure is observed. The vehicle partially fulfils the demand of customer at a failure event and the driver retrieves routing feasibility by solely executing BF trips. In such a way, the classical recourse policy causes split services at the failure events that results in customer unsatisfactory, a serious "drawback" in a managerial perspective. On the other hand, the classical policy performs the least expected number of recourse actions that can be accounted as an "advantage". Given a routing decision, to obtain a set of optimal (i.e., entailing the least expected recourse cost) recourse decisions, one can perform recourse actions using an *Optimal Restocking Policy* in the VRPSD context. The optimal restocking policy employs a dynamic programming approach to schedule BF and PR trips based on customer-specific thresholds. After serving a specific customer, if the residual capacity of the vehicle is less than a specific value i.e., *customer-specific threshold*, which is computed by the dynamic programming approach, the vehicle prescribes a PR trip. It should be noted that the optimal policy is route-dependent, thus, any changes in the routing decision makes an optimal policy generally suboptimal. Then, the optimal customer-specific thresholds cannot be modified or controlled by a transportation company through operational conventions, which is a "drawback" in a managerial perspective. In addition, there is a "lack of an exact solution approach" to optimally obtain both routing and recourse decisions. Then, designing an exact method that efficiently results in optimal decisions in the context of VRPSD is mainly of interest.

This dissertation consists of three papers, concerning the development of recourse policies for the VRPSD that are suitable in managerial settings as well as exact solution approaches to solve VRPSD instances efficiently. Dealing with unexpected events in the uncertain environments, transportation companies set *fixed operational conventions* to simplify their operations and to achieve a high level of routing consistency. Such operational conventions, varying by the type of service level which are aimed by the transportation companies, can be translated to the fixed rules for performing recourse actions. Then, our main goal is to design recourse policies which are able to prescribe a set of predetermined recourse actions based on fixed rules.

1.1 Problem Statement

In this dissertation, we model the VRPSD using the a priori optimization paradigm, by which the problem is formulated as a two-stage stochastic integer program with recourse. In this manner, overall decision making process is decomposed into two mutually exclusive stages. In the first-stage, the routing decisions are taken. Then, stochasticity reveals itself in the execution of routing decisions. The second stage consists of making recourse decisions according to a selected recourse policy, which maps a set of pre-determined recourse actions to the customers scheduled along the route. Given a recourse policy employed to perform recourse actions, solving the VRPSD exactly results in an optimal routing decision with the minimum total cost, including routing cost and associated expected recourse cost.

In practice, transportation companies, executing routing decisions on a daily basis repeatedly, need to properly select cost-effective recourse policies. Moreover, setting fixed operational rules to perform recourse actions are desired to preserve operational consistency.

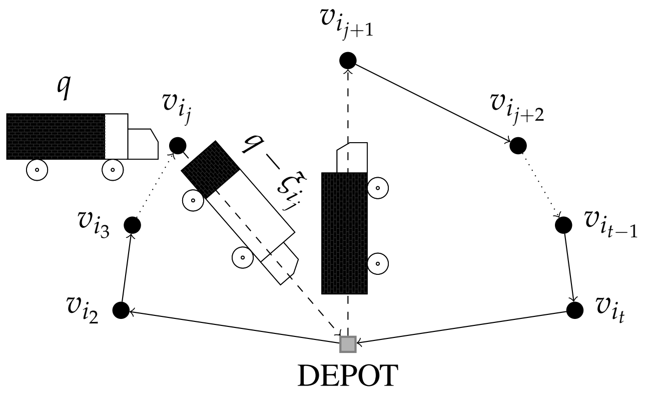
In this dissertation, we discuss and explore efficient recourse policies which do not suffer from the above-mentioned drawbacks. To avoid the drawback (e.g., service interruptions leading to customer dissatisfaction) of classical recourse policy, one can specify a minimum service level, to be provided by the decision maker, in the form of the customer-specific thresholds. In this manner, the vehicle visits a customer only if it can fulfil at least the predetermined minimum service level. To provide the service level, the vehicle must execute a PR trip, whenever it is needed. However, an important question that remains unanswered is how can proactive recourse actions be implemented in the static decision environment defined in the a priori approach? To handle this question we propose two families of recourse policy including the fixed rule-based and hybrid ones for the VRPSD. As such, static operational rules streamline the operations in a manner that greatly simplifies policies.

In the following, we first provide a general framework to classify the static operational rules that may be applied in the VRPSD context. The proposed taxonomy groups the possible policies in three general classes: (i) *volume-based policies*, (ii) *risk-based policies* and (iii) *distance-based policies*. Hybrid policies which consider several operational rules can also be envisioned in a practical setting. These policies (i.e., fixed rule-based and hybrid ones) are discussed in subsections §1.1.1 and §1.1.2. Furthermore, the VRPSD under an optimal restocking policy is also discussed in §1.1.3.

1.1.1 Volume-Based Rules

In the first problem of this dissertation, we focus on the VRPSD under the volume-based recourse policies. We are interested in controlling recourse actions based on the fixed volume-based operational conventions. It should be noted that we need to translate these conventions to the fixed operational rules which enable us implementing preventive recourse actions. The problem is how to define the considered operational rules and show how these operational rules can be employed using a fixed threshold-based policy to govern the recourse actions.

Transportation companies can then adjust the customer-specific thresholds which reflect company's operational policies allowing them to control the risk of encountering failures. To execute volume-based policies we need to define the customer-specific thresholds as a function of the capacity of the vehicle or the demands of the customers. A schematic example of a threshold-based policy is shown in Figure 1.1. As shown in Figure 1.1, the recourse policy can be implemented efficiently by a driver using a **Daily log-trip sheet**. In this manner, the vehicle after serving the j customer with q units of residual capacity less than threshold value θ_{i_j} performs a preventive return.



(a)

Daily log-trip sheet

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

(b)

Figure 1.1 – The vehicle is executing a PR trip to provide the customer-specific threshold.

1.1.2 Risk-Distance Based Rules

In the second problem, we study the VRPSD under a hybrid recourse policy. We aim to establish a unified recourse decision making process which incorporates two measures including risk of failure and distances-to-travel enabling us to prescribe PR trips. We design a risk measure which computes the risk of failure at the next customer and compares it with predetermined thresholds. Transportation companies can control the risk of failure according to these thresholds and prescribe preventive returns whenever the risk of failure is too high. When the risk of failure is neither too high for performing a PR trip nor to low for proceeding the planned route, we implement a distance-based measure. We then implement a distance measure which compares the PR trip cost at the current customer with the average cost of future BF trips. Then, we aim to integrate this hybrid recourse in an exact solution approach for the VRPSD.

1.1.3 Optimal Rules

In the third problem of this dissertation, we study the VRPSD under an optimal restocking policy. The optimal restocking recourse strategy is early proposed in 1980. Under such recourse policy, a set of optimal customer-specific thresholds are obtained such that the recourse cost will be minimized. In this setting, the vehicle after serving a specific customer with the residual capacity less than the associated threshold must perform a preventive return. The optimal restocking policy has been only integrated in the heuristic and metaheuristic solution methods which result in a pair of suboptimal routing and recourse decisions. In this part of dissertation, we aim at solving the VRPSD under an optimal restocking policy exactly.

1.1.4 Solution Method

The broadest problem investigated in this dissertation is to develop solution methodologies for tackling the VRPSD under various recourse policies proposed in this research. In this dissertation we adopt the integer L-shaped algorithm which is initially developed to solve two-stage stochastic integer programs with recourse(first-stage variables are binary). The integer L-shaped algorithm as a general branch-and-cut (B&C) procedure is then employed to solve the multi-VRPSD. As a general B&C procedure, the algorithm adds the violated constrained which are initially relaxed and achieves integrality by branching scheme. Furthermore, the expected recourse cost function is replaced by a nonnegative variable and initially bounded from below. Since the exact solution framework only can improve the overall upper bound by computing the recourse cost of integer solution, other types of valid inequalities, by which the expected recourse cost can be efficiently bounded, are introduced. Such valid inequalities enhance the efficiency of the integer L-shaped algorithm by providing various bounding schemes for fractional solutions with certain structures. In order to use these type of valid inequalities, a valid lower bound for the expected recourse cost of fractional solutions under the proposed recourse policies must be computed. To our best knowledge, computing such valid lower bounds, when customer demands are discrete, is only restricted to the single-VRPSD with restricted number of failures.

1.2 Thesis Contributions

The author of this thesis conducted the research proposed in Chapters 3, 4, and 5. He wrote the code, implemented and tested the models and the algorithms, and analysed output data. Michel Gendreau,Walter Rei, and Ola Jabali supervised the thesis, proposed the general orientation, helped with the design of the numerical experiments, discussed the results and commented on the three manuscripts at all stages.

The main contributions of this dissertation are as follows in details:

— We present a general framework to classify recourse policies.

— We introduce a new family of recourse policies as rule-based recourse policies and propose three different volume-based rules to prescribe recourse actions.

— We adapt the integer L-shaped algorithm to solve the VRPSD under rule-based recourse policies. To do so, we develop several lower bounding procedures to enhance the efficiency of the algorithm.

— We conduct an extensive computational study using a large set of randomly generated instances, to illustrate the performance of our algorithm under rule-based policies.

— We present a new family of recourse polices as mixed recourses. We define a unified decision rule consisting of a risk measure that computes the probabilities of failure at the next or following customers and a distance measure that specifies whether prescribing a PR trip is more benefited or not.

— We adapt the integer L-shaped algorithm to solve the VRPSD under a mixed policy and provide lower bounding techniques which speed up the overall branch-and-cut procedure.

— We redevelop the integer L-shaped algorithm to solve the VRPSD under an optimal restocking policy, exactly.

— To enhance the efficiency of the integer L-shaped algorithm, various lower bound improving techniques are established.

— A general lower bound which approximates the expected recourse function is determined.

Our contributions, based on how routing and recourse decisions are made, are summarized in Table 1.I and are specified by the italics.

Table 1.I – Classification of the solution approaches for the VRPSD.

	Recourse Decisions	
Routing	Static	Dynamic
Static	a priori (BF), *rule-based*(BF/PR), *hybrid*(BF/PR)	*optimal restocking*(BF/PR)
Dynamic	—-	reoptimization(BF/PR)

1.3 Thesis Organization

This dissertation addresses the VRPSD under various recourse policies. This dissertation is made of six chapters three of which correspond to articles submitted to scientific journals. Because of that Chapters 3-5 reproduce the exact format of articles that submitted. In particular, they include abstracts, separate references and appendices. Chapter 3 presents *A Rule-Based Recourse for the Vehicle Routing Problem with Stochastic Demands* which has been submitted to the Journal of *Transportation Science*. Chapter 4 presents *A Hybrid Recourse Policy for the Vehicle Routing Problem with Stochastic Demands* which has been submitted to the *EURO Journal on Transportation and Logistics*. Chapter 5 presents *An Optimal Restocking Recourse Policy for the Vehicle Routing Problem with Stochastic Demands* which has been submitted to the *European Journal of Operational Research*. We already have provided the general introduction and problem statement in the Chapter 1. The Chapter 2 covers the literature review of the VRPSD including existing modeling paradigms and solution frameworks, which have been presented in the previous works.

In Chapter 3, we present the first paper of this dissertation. This chapter addresses the problem stated in Section 1.1.1 in which we introduce a rule-based recourse policy. An exact solution methodology is then developed to solve the VRPSD under a rule-based recourse policy. Finally, the performance of the proposed methodology is investigated by an extensive numerical experiments in the same Chapter.

In Chapter 4, forming the second paper, the aim is to introduce a hybrid recourse policy as stated in Section 1.1.2. We combine two new measures including the risk of failure and the distances-to-travel into a single recourse decision rule. Moreover, an exact solution methodology is developed to solve the VRPSD under this hybrid recourse policy. We finish the chapter by presenting the numerical experiments and the evaluation of the proposed algorithm.

Chapter 5, which stands as the third paper of this dissertation, is devoted to the study of the VRPSD under an optimal restocking policy, as stated in Section 1.1.3. The integer L-shaped algorithm which enhanced by lower bounding schemes is proposed to optimally solve the VRPSD under an optimal restocking policy. The performance of the proposed method is then investigated by an extensive numerical experiments.

Finally, Chapter 6 provides conclusions as well as the future research avenues.

CHAPTER 2

LITERATURE REVIEW

The Vehicle Routing Problem (VRP), introduced in the seminal paper of Dantzig and Ramser [15], is one of the most studied combinatorial optimization problems in the field of operations research. The classical VRP aims at designing a set of vehicle-routes, with minimum travelled-cost through a set of geographically dispersed locations, which start and end at a single depot location to deliver or pickup goods. 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 [55] for a thorough exposition of the problem and its main variants).

In reality, however, routing problems have to deal with several sources of uncertainty: demands, travel and service times, customer presence, 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 bad quality solutions, see Louveaux [36]. Therefore, there is a need to develop specialized stochastic optimization paradigms that explicitly model random variables. While they have received much less attention than deterministic VRPs, SVRPs have been investigated by several authors; see Gendreau et al. [22] for a survey of the SVRP literature. In this thesis, we focus on the variant of the problem in which customer demands are uncertain, being only specified through probability distributions. In this variant which is called the Vehicle Routing Problem with Stochastic Demands (VRPSD), customer demands can be observed upon arrival at the customer location.

We will use the following notation to model the VRPSD. We denote by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ a complete undirected graph, which consists of a set of vertices denoted by $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ and a set of edges denoted by $\mathcal{E} = \{(v_i, v_j) | v_i, v_j \in \mathcal{V}, i < j\}$. We present the depot by vertex v_1, where a fleet of m identical vehicles with a capacity of Q is based. Let vertex v_i (for $i = 2, \ldots, n$) represents the i^{th} customer with stochastic demand

ξ_i, which 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 of observing the l^{th} demand level (i.e., value ξ_i^l), i.e., $\mathbb{P}[\xi_i = \xi_i^l] = p_i^l$. We define matrix $C = (c_{ij})$ as the symmetric distance matrix, in which each c_{ij} is associated to the length of $(v_i, v_j) \in \mathcal{E}$.

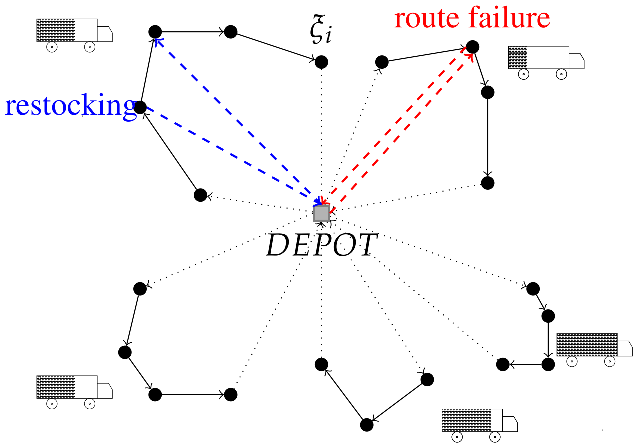


Figure 2.1 – VRPSD under a set of predefined recourse actions

In presence of demand stochasticity, a planned route may *fail* at a specific customer where the observed demand exceeds the residual capacity of the vehicle. In such a case, a *route failure* is occurred, called by Dror and Trudeau [16]. Then, a *recourse action* must be taken to preserve route feasibility, thus entailing extra costs. These recourse actions can either be reactive (i.e., to be performed once after a route failure occurred) or proactive (i.e., in anticipation of future failures taking place along the route). As a reactive recourse action, the vehicle executes a back-and-forth (BF) trip to the depot, to refill its capacity for completing the service where the failure is occurred. In the case of an *exact stockout*, in which the observed demand turns out the exact value of the residual capacity (is the case only having discrete distributions), the vehicle performs a *restocking trip*; the vehicle replenishes the capacity at the depot and then proceeds to the next customer, see Gendreau et al. [20] and Hjorring and Holt [27]. In order to maintain route feasibility, the vehicle prescribe *preventive restocking* (PR) trips, when the residual capacity becomes too low, see Yee and Golden [58] and Yang et al. [57]. PR trips are considered as proactive recourse actions, because the vehicle preventively returns to the depot before an actual failure occurs. To model the VRPSD illustrated in Figure 2.1, one must determine a unified decision rule, called *recourse policy*, which governs the prescription of a set of pre-determined recourse actions. We refer to the *modeling paradigm* in the context of the VRPSD as: how the overall decisions, routing and recourse ones, are taken through the information revelation, see Table 1.I. There are four modeling paradigms to model the VRPSD as the *a priori optimization*, *re-optimization*, *chance constraints* and *robust optimization* approaches, which are elaborated in the Sections §2.1, §2.2, §2.3 and §2.4, respectively. We then study various aspects including formulations, recourse policies, and exact and heuristic solution methods.

2.1 A Priori Optimization

In the *a priori optimization* approach, originally proposed by Dror et al. [17] and further investigated Bertsimas et al. [4], one decomposes the VRPSD decisions into two sets, where routing and recourse decisions must be taken before and after the demand realizations, respectively. In such a way, the first-stage is equivalent to find a set of a priori routes, while certain demands are not yet known. When uncertainty reveals itself in the execution of the planned route and the vehicle realizes the actual demands, the second-stage problem includes finding a set of recourse actions preserving routing feasibility with minimum expected recourse-cost. Overall, the aim in the a priori optimization approach is to find a routing decision which incurs the least total routing and expected recourse costs. In more technical terms, we model the VRPSD by an a priori approach as a two-stage Stochastic Program with Recourse (SPR).

In this section, we start with the two-stage stochastic programming with fixed recourse presented in Birge and Louveaux [9] as follows,

$$\begin{aligned} \min_{x} \quad & c^t x + \mathcal{Q}(x) \\ \text{s.t.} \quad & Ax = b, \\ & x \geq 0, \end{aligned} \tag{First-Stage}$$

where

$$\begin{aligned} \mathcal{Q}(x) &= \mathbb{E}_{\xi}\left[Q(x, \xi(\omega))\right] \\ Q(x, \xi(\omega)) &= \min_{y}\{q(\omega)^t y | Wy = h(\omega) - T(\omega)x,\ y \geq 0\} \end{aligned} \tag{Second-Stage}$$

The first-stage problem (First-Stage) is established by the decision vector $x \in \mathbb{R}^{n_1}$, the cost vector $c \in \mathbb{R}^{n_1}$, right-hand-side $b \in \mathbb{R}^{m_1}$, and technology matrix $A \in \mathbb{R}^{m_1 \times n_1}$. Moreover, $Ax = b$ and $x \geq 0$ express the constraints and restrictions in (First-Stage) which must be satisfied by each feasible decision x. Suppose that the feasible first-stage decision x is given. In the second-stage, the stochasticity $\omega \in \Omega$ becomes known. In this dissertation, we assume that the stochastic variables follow general *discrete* distributions. Although we represent a two-stage stochastic program with fixed recourse here, the source of stochasticity in the VRPSD is only in right-hand-side, customers demand, known by discrete probability distributions beforehand. The second-stage problem (Second-Stage) is established by the decision vector $y \in \mathbb{R}^{n_2}$, the cost vector $q(\omega) \in \mathbb{R}^{n_2}$, right-hand-side $h(\omega) \in \mathbb{R}^{m_2}$, and technology matrices $T(\omega) \in \mathbb{R}^{m_2 \times n_1}$ and $W \in \mathbb{R}^{m_2 \times n_2}$ associated to x and y variables, respectively. For the second-stage problem, each realization $\omega \in \Omega$ generates known vectors and matrices $q(\omega)$, $h(\omega)$, and $T(\omega)$. The stochastic vector $\xi(\omega)$ includes all stochastic sources, $q(\omega)$, $h(\omega)$, and $T_1(\omega), \ldots, T_{m_2}(\omega)$, totally with $N = n_2 + m_2 + m_2 \times n_1$ components. Then, for each realization ω the second stage consists of finding the optimal recourse/corrective decision vector y which minimizes the objective function $Q(x, \xi(\omega))$. Finally, $\mathcal{Q}(x)$ can be computed by taking expectation of $Q(x, \xi(\omega))$ with respect to ξ. We should note that ξ has a finite *support* set $\Xi \subseteq \mathbb{R}^N$, where N is the size of vector $\xi(\omega)$ such that the probability over support set Ξ is one.

One can model the VRPSD using a two-stage stochastic *integer* program with recourse (TS-SIPR) under the a priori optimization approach, see Dror et al. [17] and Gendreau et al. [22] for a detailed exposition of modeling frameworks. Based on the principles of the a priori optimization approach, the routing decisions denoted by decision vector x are taken in the first stage. In such a way, each routing decision x in the VRPSD formulation 2.1-2.7 consists of m vehicle routes, each serviced by a vehicle with Q units of capacity, where route r can be represented as follows,

$$\vec{v} = (v_1 = v_{r_1}, v_{r_2}, \ldots, v_{r_i}, v_{r_j}, \ldots, v_{r_t}, v_{r_{t+1}} = v_1). \qquad \text{(a priori route)}$$

For each pair of successive customers v_{r_i} and v_{r_j} in $\vec{v}$, x_{ij} takes one; otherwise zero. As presented in Gendreau et al. [20] and Laporte et al. [35], the TS-SIPR model for the VRPSD can be formulated as follows:

$$\begin{aligned}
\underset{x}{\text{minimize}} \quad & \sum_{i<j} c_{ij}x_{ij} + \mathcal{Q}(x) && && (2.1)\\
\text{subject to} \quad & \sum_{j=2}^{n} x_{1j} = 2m, && && (2.2)\\
& \sum_{i<k} x_{ik} + \sum_{k<j} x_{kj} = 2, && k = 2, \ldots, n && (2.3)\\
& \sum_{v_i, v_j \in S} x_{ij} \leq |S| - \left\lceil \frac{\sum_{v_i \in S} \mathbb{E}(\xi_i)}{Q} \right\rceil, && (S \subset \mathcal{V} \setminus \{v_1\}; 2 \leq |S| \leq n-2) && \\
& && && (2.4)\\
& 0 \leq x_{ij} \leq 1, && 2 \leq i < j < n && (2.5)\\
& 0 \leq x_{1j} \leq 2, && j = 2, \ldots, n && (2.6)\\
& x = (x_{ij}), && \text{integer.} && (2.7)
\end{aligned}$$

The objective function of the first-stage consists of the distances traveled by the fleet of vehicles i.e., $\sum\limits_{i<j} c_{ij}x_{ij}$. Constraints (2.2) are the degree constraints and establish m

vehicle-routes that start and end at the depot; Constraints (2.3) are the flow constraints which ensure that each customer is visited exactly once; Constraints (2.4) stand as both subtour elimination and capacity constraints, which remove both subtours, and infeasible routes with an excessive expected demand. Constraints (2.4) also implies that for the a priori route $\vec{v}$ the total expected demand of the route does not exceed the vehicle capacity. Constraints (2.5-2.7) impose the necessary bounds and integrality restrictions on the decision variables. The objective function (2.1) minimizes the total cost including the routing (i.e., first-stage) cost, and the expected recourse (i.e., second-stage) cost $\mathcal{Q}(x)$. In the a priori optimization paradigm, the vehicle realizes actual demands upon arriving at the customer location. Given the vehicle route $\vec{v}$, the vehicle executing $\vec{v}$ may fail to complete the service at a specific customer, at which the observed demand exceeds the residual capacity of the vehicle. Such an occurrence is called *route failure* and the route becomes infeasible. Then the vehicle must perform recourse/corrective actions to preserve route feasibility. The recourse actions we study in this dissertation are in the form of return trips to the depot. The second-stage problem $\mathcal{Q}(x)$ then consists of finding a set of recourse actions by means of returning trips which minimizes the expected recourse costs.

Dror and Trudeau [16] show that when customer demands are independent, $\mathcal{Q}(x)$ can be computed separately for each routes. Moreover, the expected recourse cost of the a priori route $\vec{v}$ depends on which orientation the route is executed. We then denote by $\mathcal{Q}^{k,\delta}$ the expected recourse cost of the k^{th} a priori route in the orientation δ ($\delta = 1,2$) and we have,

$$\mathcal{Q}(x) = \sum_{k=1}^{m} \min\{\mathcal{Q}^{k,1}, \mathcal{Q}^{k,2}\}. \tag{2.8}$$

The k^{th} a priori route $\vec{v}$ will be executed under a given recourse policy in both directions to obtain the minimum expected recourse cost, as presented by function (2.8). Finally, the specific recourse policy to compute $\mathcal{Q}^{k,\cdot}$ is the subject of the following Section.

2.1.1 Recourse Policies

In this part, we briefly study the various recourse strategies proposed in the literature. A recourse policy can be defined as a plan to prescribe a set of predetermined recourse actions. The set of predetermined recourse actions typically vary with applications. Two main recourse policies proposed for the VRPSD are represented in Sections §2.1.1.1 and §2.1.1.2.

2.1.1.1 Classical Recourse Policy

Dror and Trudeau in [16] propose the classical recourse function (2.9), which considers the impact of both (i) the location of a route failure, and (ii) the direction of the planned route on the expected recourse cost. The classical recourse policy consists of following the planned route until it fails due to an excessive observed demand at a specific customer. The classical recourse policy performs only BF trips along the a priori route (in the case of continuous distributions like Normal, Poisson, etc, i.e., the distributions with accumulative property [1]). For the k^{th} a priori route $\vec{v} = (v_1 = v_{i_1}, v_{i_2}, \ldots, v_{i_j}, \ldots, v_{i_t}, v_{i_{t+1}} = v_1)$ being executed by a vehicle with capacity Q, the recourse function to compute the expected recourse cost in the first direction can be presented as:

$$\begin{aligned} Q^{k,1} &= \sum_{j=2}^{t} \sum_{l=1}^{j-1} \mathbb{P}(\sum_{s=2}^{j-1} \xi_{i_s} \leq lQ < \sum_{s=2}^{j} \xi_{i_s}) 2c_{1,i_j}, \\ Q^{k,1} &= \sum_{j=2}^{t} \sum_{l=1}^{j-1} \left[F^{j-1}(lQ) - F^{j}(lQ) \right] 2c_{1,i_j}, \end{aligned} \tag{2.9}$$

where, c_{1,i_j} is the distance between the j^{th} customer and the depot and *the probability of l^{th} failure at the j^{th} customer*, denoted by $F^{j-1}(lQ) - F^{j}(lQ)$, can be computed by

[1] The sum of several identical independent and Ψ distributed random variables is also a random variable with Ψ distribution i.e., $\xi_i \sim \Psi$ then $\sum_i \xi_i \sim \Psi$

using the cumulative probability $F^j(lQ)$ as follows,

$$\begin{aligned} F^j(lQ) &= \mathbb{P}(\sum_{s=2}^{j} \xi_{i_s} \leq lQ) \\ F^{j-1}(lQ) - F^j(lQ) &= \mathbb{P}(\sum_{s=2}^{j-1} \xi_{i_s} \leq lQ < \sum_{s=2}^{j} \xi_{i_s}). \end{aligned} \tag{2.10}$$

In (2.10), the cumulative probability $F^j(lQ)$ expresses the probability of not having l^{th} failure at the j^{th} customer. Teodorović and Pavković [54] also model the recourse function (2.9) for the VRPSD in which all customer demands follow uniform distributions.

Gendreau et al. [20] tailor the classical recourse policy for the discrete demand distributions. In this setting if the observed demand and residual capacity are the same (which is called an *exact stockout*), then the vehicle executes a *restocking* trip, see also Hjorring and Holt [27]. It should be noted that the recourse policy proposed in Gendreau et al. [20] has presented for the VRP with stochastic demands and customers (VRPSDC). Therefore, the classical recourse policy when executing both BF and restocking trips can be presented as follows,

$$\begin{aligned} \mathcal{Q}^{k,1} &= \sum_{j=2}^{t} \sum_{l=1}^{j-1} \mathbb{P}(\sum_{s=2}^{j-1} \xi_{i_s} < lQ < \sum_{s=2}^{j} \xi_{i_s}) 2c_{1,i_j} \\ &+ \sum_{j=2}^{t} \sum_{l=1}^{j-1} \mathbb{P}(\sum_{s=2}^{j-1} \xi_{i_s} < lQ = \sum_{s=2}^{j} \xi_{i_s})(c_{1,i_j} + c_{1,i_{j+1}} - c_{i_j,i_{j+1}}) \end{aligned} \tag{2.11}$$

where we denote by $\mathbb{P}(\sum_{s=2}^{j-1} \xi_{i_s} < lQ = \sum_{s=2}^{j} \xi_{i_s})$ the probability of having an exact stockout at the j^{th} customer as the l^{th} replenishment decision and we denote by $c_{1,i_j} + c_{1,i_{j+1}} - c_{i_j,i_{j+1}}$ the restocking trip cost at the j^{th} customer.

2.1.1.2 Optimal Recourse Policy

An *optimal restocking policy*, as an optimal operating strategy in the context of the VRPSD, is proposed by Yee and Golden [58]. Under such a recourse policy, the vehicle prescribes PR trips as well as BF trips. In this policy, the proceeding cost-to-go (the cost incurred by proceeding the planned route to the next customer) is compared with the case that the vehicle resumes the service after performing a PR trip at the current customer. The policy uses this comparison to take optimal actions, which then minimize the expected recourse cost. The optimal restocking policy results in a set of optimal customer-specific thresholds, see Yee and Golden [58] and Yang et al. [57]. If the residual capacity after serving the customer is high enough (i.e., greater or equal to the threshold), then the vehicle proceeds directly to the next customer, otherwise, the vehicle must prescribe a PR trip. The optimal policy can be presented as follows,

$$F_{i_j}(q) = \min \begin{cases} H_{i_j,i_{j+1}}(q) : c_{i_j,i_{j+1}} + \displaystyle\sum_{l:\xi^l_{i_{j+1}} \leq q} F_{i_{j+1}}(q - \xi^l_{i_{j+1}}) p^l_{i_{j+1}} + \\ \qquad\qquad\qquad\qquad \displaystyle\sum_{l:\xi^k_{i_{j+1}} > q} [b + 2c_{1,i_{j+1}} + F_{i_{j+1}}(Q + q - \xi^l_{i_{j+1}})] p^l_{i_{j+1}}, \\ H'_{i_j,i_{j+1}}(q) : c_{1,i_j} + c_{1,i_{j+1}} + \displaystyle\sum_{l=1}^{s_{i_{j+1}}} F_{i_{j+1}}(Q - \xi^l_{i_{j+1}}) p^l_{i_{j+1}}, \end{cases} \tag{2.12}$$

where $H_{i_j,i_{j+1}}(q)$ and $H'_{i_j,i_{j+1}}(q)$ express the total costs associated to the proceeding and restocking decisions after serving the ${i_j}^{\text{th}}$ customer, respectively and $F_{i_j}(q)$ represents the optimal expected cost-to-go after serving the j^{th} customer with q units of residual capacity onboard. The first term in (2.12) expresses the expected cost of proceeding to the $j+1^{\text{th}}$ customer consisting of the routing cost from the j^{th} customer to the $j+1^{\text{th}}$ customer, the expected cost of serving $j+1^{\text{th}}$ customer, and the expected cost of failures at the $j+1^{\text{th}}$ customer. The second term in (2.12) represents the cost of PR trip from the j^{th} customer to the $j+1^{\text{th}}$ customer and serving the $j+1^{\text{th}}$ customer with full capacity of the vehicle. The optimal policy takes the optimal action between proceeding and replenishing by choosing the action which incurs the minimum optimal cost-to-go. Yang et al. [57] compute the optimal policy for a single route with up to 15 customers. It is shown that there is a single route which always is as economical as multiple routes (i.e., the execution of several routes in a row under an optimal restocking policy is at least as beneficial as executing each route under the optimal policy separately. Bianchi et al. [7] improve the computational performance presented by Yang et al. [57] by implementing hybrid metaheuristics for the VRPSD. The local search procedures (i.e., exploring the neighbors) are further improved by using the deterministic information (length of the TSP tour).

2.1.1.3 Miscellaneous Recourse Policies

There are a few more recourse policies that are proposed in the literature. Dror and Trudeau [16] present an adaptation of the classical recourse policy, in which after the occurrence of the first failure, all unvisited customers are penalized by BF trips. This recourse policy highly penalizes route failures. A paired locally coordinated recourse strategy is proposed by Ak and Erera [1]. In this recourse strategy, a set of a priori routes are generated in the first stage and then are paired. In each pair, one a priori tour is labeled as type (I), and the other is labeled as type (II) (the a priori tours that are not paired will be considered as type (II)). When the vehicle executing a type I fails, it leaves the unvisited customers to the vehicle performing the paired route, type II, which is set to perform BF trips. Chepuri and Homem-De-Mello [12] propose a recourse strategy in which the vehicle takes no action after the first failure. Then, the route will be penalized with a penalization of customers at which the service is incomplete as well as unvisited customers. Juan et al. [29] introduce a safety stock approach for the the VRPSD. In this approach, a certain amount of surplus vehicle capacity is considered as a safety stock or buffer to be used in failure events.

2.1.2 Solution Methods

Various approaches have been proposed to solve the VRPSD. Some of these approaches are developed to solve the deterministic counterparts of the VRPSD (e.g., chance-constrained programming (CCP) models or robust counterpart of the VRPSD which both are elaborated latter) that are not the subject of this review. We then restrict ourselves to the VRPSD which are modeled through stochastic optimization models (in this section, TS-SIPR). Here, we try to cover almost all of the solution methods that are proposed to tackle the VRPSD. The presented dichotomy is further portrayed between exact methods on one side and heuristic and metaheuristic frameworks on the other side.

2.1.2.1 Exact Solution Methods: Integer L-shaped Algorithm

The Integer *L*-shaped algorithm is devised by Laporte and Louveaux [34] to solve stochastic integer programs with complete recourse. The Integer *L*-shaped algorithm also stands as an integer extension of the *L*-shaped algorithm of Van Slyke and Wets [56] for continuous variables. The *L*-shaped algorithm of Van Slyke and Wets [56] is a Benders decomposition methodology (Benders [3]) tailored for stochastic programs with the L-shaped form. Here, we use the notation used in Laporte and Louveaux [34] to present the TS-SIPR as follows

$$\underset{x}{\text{minimize}} \qquad c^t x + Q(x) \tag{2.13}$$

$$\text{subject to} \qquad Ax = b, \tag{2.14}$$

$$x \in X \tag{2.15}$$

where, $Q(x)$ in objective function (2.13) expresses the second-stage problem, and Constraints $x \in X$ consists of various *complex* constraints $Dx \geq d$ and other restrictions (e.g., in the VRPSD stands also for subtour elimination constraints) as well as integrality requirements. In the Integer *L*-shaped algorithm as branch-and-cut (B&C) procedure, a master problem here denoted by *current problem* (CP) is initially established by relaxing complex and integrality requirements $x \in X$, and replacing the expected recourse function $\mathcal{Q}(x)$ with Θ, which will be bounded from below by a general lower bound *L*. Then, CP^0 can be presented by (2.16) such that (2.14), (2.20), and (2.21) hold. For a given stage of the algorithm, we consider the following model,

$$
\begin{aligned}
\text{CP : } \underset{x,\Theta}{\text{minimize}} \quad & c^t x + \Theta && && (2.16)\\
\text{subject to} \quad & Ax = b, && && (2.14)\\
& D_k x \geq d_k, && k = 1, \ldots, s && (2.17)\\
& D'_k x + \Theta \geq d'_k, && k = 1, \ldots, s' && (2.18)\\
& E_k x + \Theta \geq e_k, && k = 1, \ldots, t && (2.19)\\
& \Theta \geq L, && && (2.20)\\
& x \text{ continuous} && && (2.21)
\end{aligned}
$$

Feasibility constraints (2.17) in the form of $D_k x \geq d_k$ will be added to the relaxation (because they are relaxed in CP^0) to preserve the feasibility requirements (in the VRPSD context, constraints (2.17) stands for subtour elimination constraints (2.4)). The algorithm adds constraints (2.19), called optimality cuts, as an extra step added to the overall B&C procedure. An optimality cut removes the integer(integrality requirement is derived by branching) solution with an excessive expected recourse cost. Gendreau et al. [20] propose a type of optimality cuts (2.19) which also bound the expected recourse cost for multi-VRPSD in the form of,

$$\Theta \geq \Theta_r \left(\sum_{ij \in S_r} x_{ij} - (|S_r| - 1) \right), \tag{2.22}$$

where, S_r represents the set of active arcs in the r^{th} integer feasible solution in the multi-VRPSD, Θ_r represents the expected recourse cost of current integer feasible solution, and a planned route in the multi-VRPSD consists of $|S_r| = n - m - 1$ arcs (n is the number of arcs and m is the number of vehicles). Laporte and Louveaux [33] propose an alternative optimality cut (2.23) to avoid numerical problems reported by Séguin [50],

$$\sum_{\substack{1 \leq i \leq j \\ x^r_{ij} = 1}} x_{ij} \leq \sum_{1 \leq i \leq j} x^r_{ij} - 1, \tag{2.23}$$

where the left-hand-side of (2.23) consists of a summation of all nonzero variables in the current solution and the right-hand-side is a constant one less than the value of the nonzero variables in the current solution. Therefore, the current solution does not satisfy in optimality cuts (2.23) and will be removed. Optimality cuts (2.23) removes an integer feasible point with an excessive expected recourse cost, while the subtree of this integer solution may contain better solutions(the subproblem obtained by adding the optimality cut will be added to the pendant subproblems to search for such solution if any exists). We note that optimality cuts are active only for one integer solution. Various valid inequalities as constraints (2.18) are being added to the CP relaxations in order to enhance overall B&C procedure (lower bounding functionals (LBFs) stands for these constraints in this context). Constraints (2.18) will be added to the fractional solutions in order to approximate the expected recourse cost from below. These valid inequalities improve the efficiency of the L-shaped method, since they are active in more than one feasible integer solution, compared with optimality cuts.

Adding lower bounding functionals is proposed by Hjorring and Holt [27] through introducing the concept of *partial routes* for the single-VRPSD. The original optimality cuts of Laporte and Louveaux [34] are modified by Hjorring and Holt [27] as,

$$\Theta \geq (\Theta_r - L)\left((\sum_{ij\in S_r} x_{ij} - (n-1))/2 \right) + L,\ \forall r = 1, \dots, R \tag{2.24}$$

where, S_r consists of indices of r^{th} routing decision and $|S_r| = n+1$ in the single-VRPSD and Θ_r presents the expected recourse cost of the planned route r[2]. Then, a *partial route* p is defined as presented in Figure 2.2. Each partial route h consists of sequenced and unsequenced set of vertices. Two sets of sequenced vertices called $S_h = \{v_1, \dots, v_{s_h}\}$ and $T_h = \{v_1, \dots, v_{t_h}\}$, and the unsequenced set of vertices called U_h are defined such that $S_h \cap U_h = \{v_{s_h}\}$, $T_h \cap U_h = \{v_{t_h}\}$, and $S_h \cap T_h = \{v_1\}$.

[2] for justification: we know that $\sum_{ij\in S_r} x_{ij} = n+1$ then $(\sum_{ij\in S_r} x_{ij} - (n-1))/2 = 1$. Therefore, (2.24) reduces to $\Theta \geq \Theta_r$ only for the route r and it reduces to $\Theta \geq L$ otherwise.

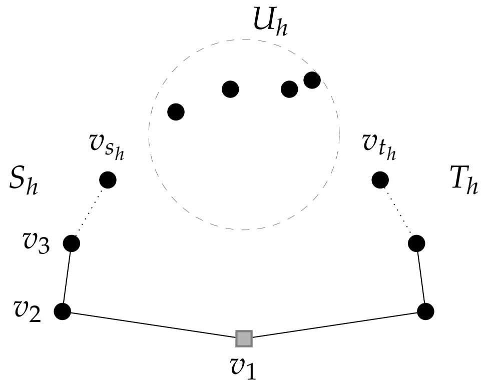


Figure 2.2 – Concept of partial route presented by [27].

Then, for partial route p, LBF cuts in the form of (2.25) are proposed by Hjorring and Holt [27] as,

$$\Theta \geq (\Theta_p - L) \left(\sum_{ij \in S_p} x_{ij} - (|S_p| - 1) \right) + L, \tag{2.25}$$

where S_p is the set of indices associated to the sequenced customers, i.e., the members of S_h and T_h and $\sum_{ij \in S_p} x_{ij} = |S_p|$. Then, the LBF cuts (2.25) sets a valid lower bound $\Theta \geq \Theta_p$ for all integer feasible solutions which can potentially emerged from partial route h by sequencing the customers in unstructured set U_h.

Constraint (2.21) sets an initial lower bound for the expected recourse cost, where L is defined by

$$L \leq \min_x \{ \mathcal{Q}(x) | (2.14), (2.15) \} .$$

Using the general lower bound L, we initially bound $\mathcal{Q}(x)$ from CP^0 enhancing the efficiency of the algorithm by tightening the optimality gap. Using the fact that one can construct m artificial clusters in the multi-VRPSD in such a way that the m nearest customers to the depot are associated to each cluster, the expected cost under the classical recourse over such clustering presents a valid general lower bound L, presented by Laporte and Louveaux [33]. An improved version of this bound is presented by Laporte et al. [35]. Overall, the computation of L is only presented for the multi-VRPSD under classical recourse, where the demand distribution are normal or Poisson. The Integer

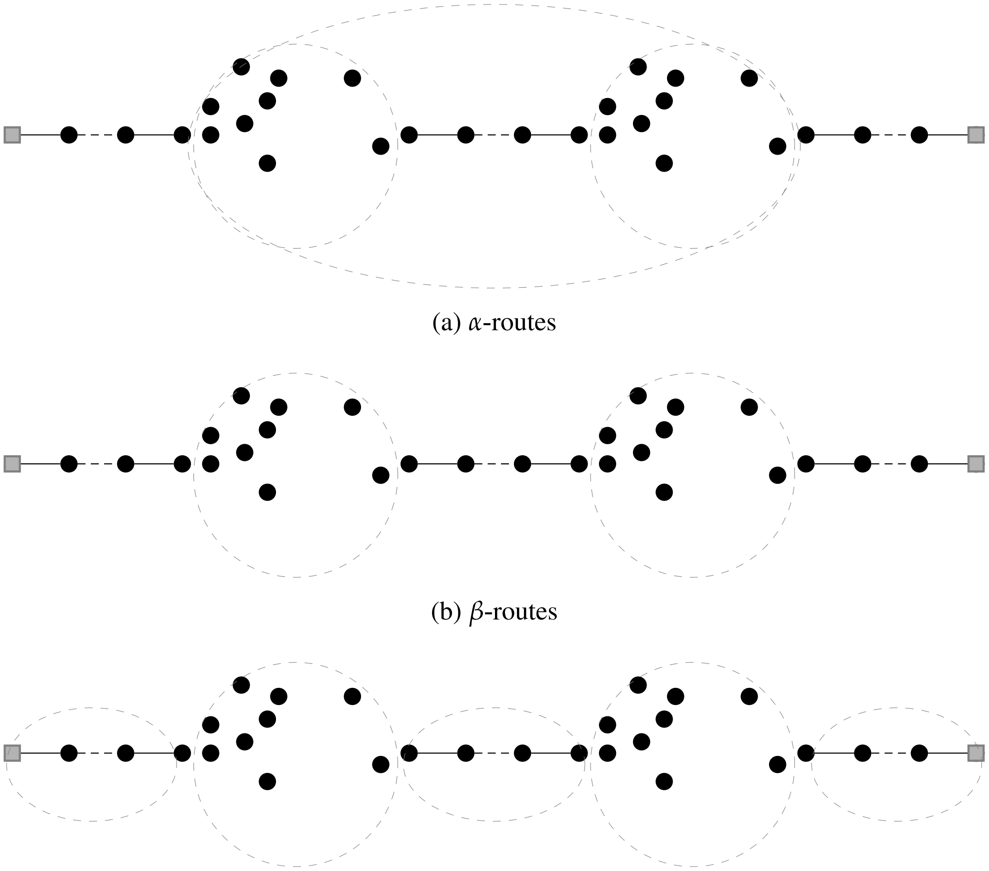
L-shaped algorithm can be expressed through the following pseudocode.

Integer L-shaped Algorithm

```
Step 1. Select one pendant node from the list pendant node(a list of relaxation problems(i.e., $CP^{\nu}$) which are not explored yet); if none exists, stop.
Step 2. Set $\nu := \nu + 1$; solve the current problem. If the current problem has no feasible solution, fathom the current node; go to Step 1. Otherwise, let $(x^{\nu}, \Theta^{\nu})$ be an optimal solution.
Step 3. Check for any relaxed constraint violation. If one exists, add feasibility cut (2.17) and update $s := s + 1$), add valid inequalities (2.18) and update $s' := s' + 1$), and return to Step 2. If $cx^{\nu} + \Theta^{\nu} > \bar{z}$, fathom the current problem and return to Step 1.
Step 4. Check for integrality restrictions. If one is violated, create two new branches following the usual branch and cut procedure; append the new nodes to the list of pendant nodes; return to Step 1.
Step 5. Compute $Q(x^{\nu})$ and $z^{\nu} = cx^{\nu} + Q(x^{\nu})$. If $z^{\nu} < \bar{z}$, then update $\bar{z}$.
Step 6. If $\Theta^{\nu} \geq Q(x^{\nu})$, then fathom the current node and return to Step 1. Otherwise impose one optimality cut (2.19), update $t := t + 1$ and return to Step 2.
```

In this algorithm, k', h' and f' are the active index sets if one exists to generate new Constraints (2.17), (2.18) and (2.19).

The concept of partial routes is further developed by Jabali et al. [28] by generalizing various structures as shown in Figure 2.3. In such a way, the traditional structure proposed by Hjorring and Holt [27] denoted by α topology in Figure 3.3a is extended to β and γ topologies as shown in Figures 3.3b and 3.3c. In this setting, a partial route with β topology consists of several sequenced and unsequenced structures, alternatively. Also, a partial route with γ topology consists of several unstructured components.



(a) α-routes

(b) β-routes

(c) γ-routes

Figure 2.3 – General topologies for partial routes proposed by Jabali et al. [28].

The LBF cuts is then proposed (2.26)

$$\Theta \geq (\Theta_p - L)\left(\sum_{h=1}^{r} W_h(x) - (r-1)\right) + L, \tag{2.26}$$

where, the new lower bounding functional $W_h(x)$ consists of both variables associated to the sequenced and unsequenced sets, see Jabali et al. [28] for more details about the definition of $W_h(x)$ and LBF cuts (2.26). We should note that Θ_p and L are computed by proposed principles of Laporte et al. [35]. Using LBF cuts (2.26) the efficiency of the Integer L-shaped algorithm is improved by providing several bounding procedures to bound the expected recourse cost at fractional solutions which represent α, β, and γ topologies.

Finally, we note that the use of LBF cuts to solve the VRPSD in which customer demands follow discrete distributions are only restricted to the single-VRPSD with restricted number of failures as proposed initially by Hjorring and Holt [27].

2.1.2.2 Exact Solution Methods: Column Generation-Based Algorithm

Dantzig-Wolfe decomposition is applied to partition VRPSD into a set partitioning master problem and a shortest path subproblem by Christiansen and Lysgaard [13] as follows,

$$\begin{aligned} \text{minimize} \quad & \sum_r c_r x_r \\ \text{subject to} \quad & \sum_r \quad \alpha_{ir} x_r = 1, \qquad \forall i \in V \setminus \{v_1\} \\ & \qquad x_r \in \{0,1\}, \quad r\text{: elementary route} \end{aligned} \tag{2.27}$$

where, α_{ir} takes value 1 only if the i^{th} customer is sequenced on the r^{th} route and c_r express the total expected costs of the r^{th} route. The pricing subproblem is established by constructing an extended graph as follows: (i) Q copies of each customer are generated to present the range of demand realizations for each customer, (ii) Q copies of the depot location where the routes end and a single depot location where all routes start are generated, and (iii) given an origin customer all arcs provided that the cumulative demand at the destination customer is less than and equal Q, are added and all paths will end at the depot location with their cumulative demand. Then, the subproblem is modeled as a shortest path problem. The auxiliary graph of subproblem is represented by Figure 2.4. All arcs, with the property that the expected demand of customers as the origin and destination of the arc is less than vehicle capacity, are added. Moreover, the subproblems can be expressed as a shortest path problem through paths which satisfy the capacity constraint. The authors show that the probability of failure at a specific customer only depends on the cumulative demand and is independent of distributed demands on the route up to the customer. By doing so, an adopted version of classical recourse which only prescribes BF trips is employed to compute the cost of BF trips. Gauvin et al. [19] use the most recent techniques for solving the column generation subproblem and improve the result presented by Christiansen and Lysgaard [13], by means of optimally solved instances and computational time.

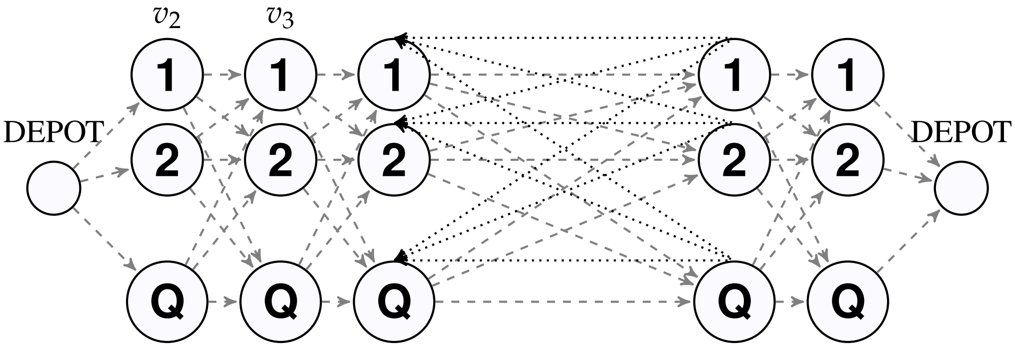


Figure 2.4 – The graph established for subproblem

2.1.2.3 Heuristic and Metaheuristic Solution Methodologies

In this section, we study heuristic and metaheuristic procedures proposed for the VRPSD. Since heuristic and metaheuristic procedures apply various improvement searches through neighbors of current solution, the major task in these solution methods is repeatedly computing the expected recourse cost. However, the evaluation of an intractable objective function turns the overall scheme inefficient, see Campbell and Thomas [10] for more details discussing challenges and advances in the a priori routing. To overcome such intractability, approximations and proxies can be employed to search for improvements, resulting in suboptimality.

Stewart and Golden [52] partition their proposed solution methodologies into two categories, where either the VRPSD is reduced to deterministic VRP (in the case of CCP models) and existing VRP heuristics are used, or the VRP heuristics (e.g., Clark-Wright saving heuristic [14]) are adapted as (2.28) to solve the VRPSD (in the case of Penalty models). Numerical experiments are conducted on multi-VRPSD with 50-75 customers and adapted savings algorithm is compared with a solution methodology based on the Lagrangian relaxation.

$$s_{ij} = c_{0i} + c_{0j} - c_{ij} + \lambda P_i + \lambda P_j - \lambda P_{ij}. \tag{2.28}$$

Dror and Trudeau [16] introduce an exact computation to compute the expected recourse cost of an a priori route under the classical recourse. The authors then propose a modified version of Clark-Wright saving heuristic in which the traditional routing cost is replaced by expected recourse cost of the route that is exactly computed. Teodorović and Pavković [54] present a Simulated Annealing technique to solve the VRPSD under classical recourse, in which the demand of customers follows uniform distribution. Then, the cumulative demand distribution is approximated by normal distribution to easily compute the probability of failure.

Gendreau et al. [21] present a tabu search heuristic TABUSTOCH for the VRPSDC. In this tabu algorithm, a penalized objective function which penalizes infeasible routes (i.e., a solution that does not contain exactly m routes) to evaluate moves in tabu search is modeled. The original objective function is replaced by an easily computed proxy to evaluate a set of neighbors. A set of neighbours are generated based on displacement of randomly selected customers for each solution. Finally, the random tabu durations are used to restrict the reinsertion or displacement of each vertex.

Yang et al. [57] present two local search heuristics using the well-known route-first cluster-second and cluster-first route-second algorithms to solve the VRPSD. In the first approach, a routing decision through all of the customers is initially made, and then the computation of expected recourse cost dynamically partitions into several clusters based an upper bound on expected recourse cost. In the second approach, a circle covering method is used to find potential seed points; next the seed points will be ranked by their distances from the depot location; finally an approximated insertion cost is used to find routing through the best clusters. The feasible solutions obtained by above-mentioned procedures are further improved by repeatedly using inter-route and intra-route exchanges.

Bianchi et al. [8] compare several metaheuristic solution approaches to solve VRPSD. To search through feasible set, the authors proposed two approximation methods to evaluate the cost of removing and reinserting a sequence of customers, providing new neighbors, in order to improve the current solution. In the first method which is based on the computation of expected cost in the VRPSD, an approximated cost of a remove-reinsert move, considering the recourse cost, is computed (e.g., see Yang et al. [57]). In the second method, the total approximated cost of a remove-reinsert move can be easily computed by the exact value of such moves, which are used in the case of the TSP. Finally, a randomized farthest insertion algorithm is used to compare the performance of proposed heuristics. Overall, proposed metaheuristics present better performance in comparison to local searches of Yang et al. [57], by resolving computational burden reported in the latter paper.

Rei et al. [43] propose local branching solution technique for the single-VRPSD. In this solution technique, an integer number to present the Hamming distance from a reference solution, which can potentially be the optimal solution of CP, is used to partition the solution space into two subregions namely (i) *left* that contains feasible solutions that present a distance equal or less than the given, and (ii) *right* that contains feasible solutions further than the given distance. Then, this theory is used to alternate the concept of partial routes proposed by [27] which bounds the recourse costs from below.

Rei et al. [44] present a hybrid Monte Carlo local branching technique to solve the single-VRPSD. Such a hybrid heuristic evaluates several given subproblems and their associated solutions to obtain an approximated new incumbent using Monte Carlo sampling approach. The new incumbent is then used in the local branching procedure to recursively generate above-mentioned subproblems.

2.2 Reoptimization Approach

Dror et al. [17] in their seminal paper propose an alternative modeling framework to model the VRPSD. The authors model the VRPSD as a Markov Decision Process (MDP) in which the state of the system is defined by vector $s = (j, q, d_{i_2}, \ldots, d_{i_j}, \ldots, d_{i_t})$, where $j \in \{v_{i_1}, v_{i_2}, \ldots, v_{i_t}\}$ expresses the current location of the vehicle, $q = 0, 1, \ldots, Q$ represent the residual capacity of the vehicle before starting the service at the location j, and values $d_{i_j} \in \{?, 0, 1, \ldots, Q\}$ for all $j = 2, \ldots, t$ express the demand of j^{th} customer that is not fulfilled yet and ? represents the state of unvisited customers. The initial state is at the depot at which all customers are unvisited, and the final state is also at the depot location at which all demand values d_{i_j} are at zero level (i.e., all of the demands are met). No solution method is provided by the authors.

The MDP modeling approach further investigated by Secomandi [48], Secomandi and Margot [49] and Novoa and Storer [42]. In this manner, the VRPSD is modeled as a stochastic shortest path problem (SSPP), in which the states are being absorbed at a terminal state with no cost (here, returning to the depot after service completion). The objective is to find a set of optimal controls, each consisting of a pair of customer and action which determines optimally which customer will be the next customer in the system and what action needs to be selected for this transition either proceed directly, or by a replenishment decision, to minimize the overall costs. In this setting, the state space of the problem is intractable. Then, the VRPSD modeled as SSPP is investigated in single vehicle case. Rollout algorithm to approximate the optimal cost-to-go functions is then employed to approximate optimal controls. Recently, Secomandi and Margot [49] further improve the rollout policy using a partial reoptimization technique, in which the size of state space is profoundly reduced by implying pre-defined partial orderings through the set of customers.

2.3 Chance Constraint Programming

Chance-Constrained Programming (CCP) approach is proposed by Charnes and Cooper [11] to tackle Stochastic Programming (SP). In this approach, a set of *probabilistic* or *chance* constraints are set to deal with stochasticity, by restricting the probability of stochastic events to be less than a preset value. Such probabilistic constraints will be transformed to deterministic counterparts.

Stewart and Golden [52] examine a CCP model for the VRPSD shown in the CCP model presented below(here, we present the model for the single-VRPSD case). In this CCP model, $\sum_{i,j} \xi_i x_{ij}$ expresses the distribution of demand for a single route in the single-VRPSD and c_{ij} represents the travelling cost between customer v_i and customer v_j. Then, the probabilistic constraint restricts optimization over feasible routes with a failure probability less than α. Under certain assumptions, i.e., (i) the stochastic demands ξ_i are independent; (ii) ξ_i's have accumulative property; (iii) the stochastic demands ξ_i have the same ratio of $\frac{\sigma_i^2}{\mu_i} = z$, the following CCP model can be transformed to a deterministic counterpart, which then can be solved using existing solution methods tailored for deterministic problems (e.g., see Dror et al. [17] for more details in translating the CCP model to its deterministic counterparts). Using Theorem 1 on page 378 of Stewart and Golden [52], the probabilistic constraints can be replaced by the nonlinear deterministic counterpart $\sum_{i,j} \mu_i x_{ij} + \tau(\sum_{i,j} \sigma_i^2 x_{ij}^2)^{1/2} \leq Q$ in which $\mathbb{P}[\frac{\sum_{i,j} \xi_i x_{ij} - \sum_{i,j} \mu_i x_{ij}}{(\sum_{i,j} \sigma_i^2 x_{ij}^2)^{1/2}} \leq \tau] = 1 - \alpha$. Finally, this constraints can be replaced by $\sum_{i,j} \mu_i x_{ij} \leq \bar{Q}$ where $\bar{Q} = [2Q + \tau^2 z - (\tau^4 z^2 + 4Q\tau^2 z)^{1/2}]/2$.

$$\begin{aligned} \text{minimize} \quad & \sum_{i,j} c_{ij}\ x_{ij} \\ \text{subject to} \quad \mathbb{P}(& \sum_{i,j} \xi_i\ \ x_{ij} \leq Q) \geq 1 - \alpha, \\ & \boldsymbol{x} = [x_{ij}] \in S_{TSP}, \end{aligned} \tag{CCP}$$

Stewart and Golden [52] also propose two penalized models in which the probabilistic constraint is relaxed as follows,

$$\begin{aligned} \text{minimize} \quad & \sum_{ij} c_{ij}\ x_{ij} + \texttt{Penalty} \\ \text{subject to} \quad & \boldsymbol{x} = [x_{ij}] \in S_{TSP}, \end{aligned} \tag{Penalty}$$

In the first model, the objective function is penalized by an expected failure cost using fixed penalty coefficient λ,

$$\texttt{Penalty} = \lambda \mathbb{P}(\sum_{i,j} \xi_i\ x_{ij} > Q).$$

In the second model, an expectation of expected excess is used to penalize the objective function by

$$\texttt{Penalty} = \sum_{l>0} l \mathbb{P}(\sum_{i,j} \xi_i\ x_{ij} - Q = l),$$

where l stands for excess values.

Overall, the CCP approaches translate the VRPSD to the approximated deterministic counterparts. In such a way, the proper extra costs associated to the failure events and the location of failure are being neglected. Dror et al. [17] restrict the occurrence of route failures in a different fashion. The VRPD is modeled as follows,

$$\begin{aligned} \text{minimize} \quad & \sum_{i,j} c_{ij} \quad x_{ij} \\ \text{subject to} \quad & \sum_{i\in S, j\notin S} x_{ij} \geq V_\alpha(S), \quad S \subseteq V,\ |S| \geq 2 \\ & \boldsymbol{x} = \quad [x_{ij}] \in S_{TSP}, \end{aligned} \tag{CCP2}$$

where, $V_\alpha(S)$ is the minimum number of vehicles needed to complete the service customers of S such that the probability of route failure in S does not exceed α, or, satisfying following condition,

$$\mathbb{P}(\sum_{i\in S} \xi_i > V_\alpha(S)Q) \leq \alpha, \tag{2.29}$$

and $V_\alpha(S) = \lceil \frac{z_\alpha(\sum_{i\in S}\sigma_i^2)^{1/2} + \sum_{i\in S}\mu_i}{Q} \rceil$ in which, z_α is the order α fractile of the standard normal distribution of the $\sum_{i\in S}\xi_i$ and each ξ_i follow normal distribution. All models presented in this section can be solved by means of solution methods which are developed to tackle the deterministic VRP.

2.4 Robust Optimization Approach

The robust optimization is an alternative framework to model uncertainty in the uncertain linear optimization problems defined by Ben-Tal et al. [2] (we use the same notations presented by authors) as,

$$\{\min_x \{c^T x + d | Ax \leq b\}\}_{(c;d;A;b)\in\mathcal{U}}.$$

In this setting, the goal is to minimize the $\min_x\{c^T x + d|Ax \leq b\}$ such that $(c;d;A;b) \in \mathcal{U}$, where $\mathcal{U}$ expresses the uncertainty set. We assume that the uncertainty set $\mathcal{U}$ can be parametrized in an affine fashion as follows,

$$\mathcal{U} = \left\{ \left[\begin{array}{c|c} c^T & d \\ \hline A & b \end{array} \right] = \left[\begin{array}{c|c} c_0^T & d_0 \\ \hline A_0 & b_0 \end{array} \right] + \sum_{l=1}^{L} z_l \left[\begin{array}{c|c} c_l^T & d_l \\ \hline A_l & b_l \end{array} \right] : z \in \mathcal{Z} \right\}$$

A robust feasible solution x is a solution that fulfills all copies as $Ax \leq b\, \forall (c; d; A; b) \in \mathcal{U}$. Given a decision vector x, the robust value $\hat{c}(x)$ of the objective function in the uncertain linear optimization is defined as $\hat{c}(x) = \underset{(c;d;A;b)\in\mathcal{U}}{sup} \{c^T x + d\}$.

Then, the robust optimization counterpart of a linear optimization problem can be established by presenting a collection (i.e., multiple copies) of the original uncertain linear optimization problem, while each copy is associated to a realization of stochastic parameters varying in a given bounded uncertain data set presented by,

$$\underset{x}{min} \left\{ \hat{c}(x) = \underset{(c;d;A;b)\in\mathcal{U}}{sup} \{c^T x + d\} : Ax \leq b\, \forall (c; d; A; b) \in \mathcal{U} \right\}$$

Equivalently, the robust version of the two-stage stochastic program presented in the section §2.1 can be presented as follows,

$$\begin{aligned} \min_{x} \max_{\xi \in \Xi} \quad & c^t x + \mathcal{Q}(x, \xi) \\ \text{s.t.} \quad & Ax = b, \\ & x \geq 0, \end{aligned} \qquad \text{(Robust Two-Stage Stochastic Program)}$$

The only existing research that studies the VRPSD modeled by robust optimization approach is presented by Sungur et al. [53]. In this modeling approach the authors first replace the subtour elimination constraints (2.4) by the Miller-Tucker-Zemlin (MTZ) version (Miller et al. [41]). Then, MTZ constraints will be replaced by a bounded uncertainty data set. Sungur et al. [53] then show that the resulted model can be reduced to the model only considering the MTZ capacity constraints with the worst-case demand realizations. Since the latter model results in the a priori routes that never fail, no recourse cost function is formulated.

Gounaris et al. [25] also tackle the robust VRP (RVRP) under demand uncertainty.

The authors develop robust rounded capacity inequalities which can be efficiently separated using a separation procedure tailored for two classes of demand distributions. An exact B&C procedure employs the mentioned separation procedure to solve RVRP efficiently.

Solano Charris [51] studies the RVRP with uncertain travel costs and bi-objective RVRP with uncertain demands and travel times. Genetic algorithm and local search-based metaheuristics are developed to tackle the first problem. Then, the genetic algorithms developed for the first problem are adapted to solve the second problem.

CHAPTER 3

ARTICLE 1: A RULE-BASED RECOURSE FOR THE VEHICLE ROUTING PROBLEM WITH STOCHASTIC DEMANDS

Chapter notes: This chapter has been submitted to the Journal of *Transportation Science*. Preliminary work was presented at the following conferences:

- ODYSSEUS 2015, Ajaccio, France, May 31th - June 5th 2015
- CMS 2015, Prague, Czech Republic, May 27-29, 2015
- Optimization Days 2014, Montreal, Canada, May 5-7, 2014

Contribution:

- The general orientations of paper are proposed by the supervisors.
- Conducting the research including the modelling of ideas, implementations and coding parts is carried out by student. The draft versions of the three papers are written by student and then are modified by supervisors.

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

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

[1]Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport (CIRRELT), C.P. 6128, succ. Centre-ville, Montréal, Québec H3C 3J7, Canada
[2]Département d'informatique et de recherche operationnelle, Université de Montréal, C.P. 6128, succ. Centre-ville, Montréal, Québec H3C 3J7, Canada
[3]Département de mathématiques et de génie industriel, Polytechnique de Montréal, C.P. 6079, succ. Centre-ville, Montréal, Québec H3C 3J7, Canada
[4]Dipartimento di elettronica, informazione e bioingegneria, Politecnico di Milano, Piazza Leonardo da Vinci, 32, Milano 20133, Italy
[5]Département de management et technologie, École des sciences de la gestion, Université du Québec à Montréal, C.P. 8888, succ. Centre-ville, Montréal, Québec H3C 3P8, Canada

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

3.1 Introduction

Since the seminal paper of Dantzig and Ramser [15], 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 [55] 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 [36]. 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. [22] 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. [22] for a thorough discussion of these paradigms). In this paper, we adopt the *a priori optimization* paradigm, originally proposed by Bertsimas et al. [4]. 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 [16]. To recover route feasibility, recourse actions must be taken. As presented in Gendreau et al. [22], 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. [20] and Hjorring and Holt [27]. 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 [58] and Yang et al. [57]. 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 **?**]), 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. [35], Christiansen and Lysgaard [13], Gauvin et al. [19], and Jabali et al. [28]) and heuristics (e.g., Gendreau et al. [21], Rei et al. [44], and Mendoza and Villegas [39]) to solve the resulting model. As an alternative to the classical recourse policy for the VRPSD, Yang et al. [57] 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. [57], 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. [5], Yang et al. [57] and Bianchi [6].

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 [48], Novoa and Storer [42] and Secomandi and Margot [49]. 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. [17], or it can be modelled as a stochastic shortest path problem, as detailed in Secomandi [47].

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 3.2.

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. [19] and Jabali et al. [28] 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 §3.2 discusses general motivations for using rule-based policies in the context of VRPSD. Section §3.3 lays out the model using a rule-based recourse, then three volume-based rules are defined. Section §3.4 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 the proposed algorithm. Section §3.5 is dedicated to numerical results and compares rule-based policies in different aspects. Section §3.6 summarizes the contribution of the paper and points out some future work.

3.2 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 [30], 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. [57], 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 meta-heuristics that involve the use of an approximation cost function to evaluate the solutions. In the case of Yang et al. [57] two heuristics were proposed for the VRPSD with PR trips.

The numerical tests performed in Bianchi [6] 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. [44] 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. [44]). 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.

3.3 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 3.3.1). 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 3.3.2). Finally, we introduce a general class of volume-based recourse policies for the VRPSD (Subsection 3.3.3).

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. [35], 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. [35]):

$$\underset{x}{\text{minimize}} \quad \sum_{i<j} c_{ij} x_{ij} + \mathcal{Q}(x) \tag{3.1}$$

$$\text{subject to} \quad \sum_{j=2}^{n} x_{1j} = 2m, \tag{3.2}$$

$$\sum_{i<k} x_{ik} + \sum_{k<j} x_{kj} = 2, \qquad k = 2, \ldots, n \tag{3.3}$$

$$\sum_{v_i, v_j \in S} x_{ij} \leq |S| - \left\lceil \frac{\sum_{v_i \in S} \mathbb{E}(\xi_i)}{Q} \right\rceil, \qquad (S \subset \mathcal{V} \setminus \{v_1\}; 2 \leq |S| \leq n-2) \tag{3.4}$$

$$0 \leq x_{ij} \leq 1, \qquad 2 \leq i < j < n \tag{3.5}$$

$$0 \leq x_{1j} \leq 2, \qquad j = 2, \ldots, n \tag{3.6}$$

$$x = (x_{ij}), \qquad \text{integer} \tag{3.7}$$

where,

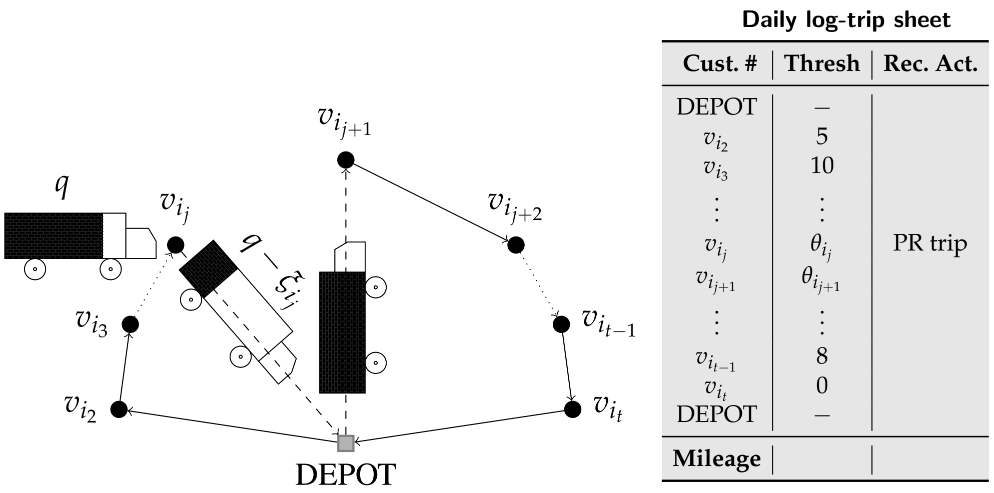
$$\mathcal{Q}(x) = \sum_{k=1}^{m} \min\{\mathcal{Q}^{k,1}, \mathcal{Q}^{k,2}\}. \tag{3.8}$$

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 [16], 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 (4.8), each route is evaluated using the two orientations and the one that minimizes the expected recourse cost is chosen. The specific computation of $Q^{k,\rho}$ will be the subject of Subsection 3.3.2.

As for the overall formulation, the objective function (3.1) 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 (3.2) and (3.3) define the structure of the a priori routes: each route starts and ends at the depot and each customer must be visited once. Inequalities (3.4) 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 (3.5), (3.6) and (3.7) impose the necessary bounds and integrality restrictions on the decision variables.

3.3.2 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 3.1. 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 3.1. 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).



(a)

Daily log-trip sheet

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

(b)

Figure 3.1 – The vehicle is executing a PR trip to provide the customer-specific threshold.

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. [57].

We now develop the recourse function that is used in model (3.1)-(3.7). 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 $j = 1, \ldots, t+1$) 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:

$$F_{i_j}(q) = \begin{cases} F_{i_j+1}(q) & \text{if } j = 1 \\ \displaystyle\sum_{s:\xi^s_{i_j}>q} \Big(b + 2c_{1i_j} + F_{i_j+1}(Q + q - \xi^s_{i_j})\Big)\, p^s_{i_j} + & \\ \displaystyle\sum_{s:q-\theta_{i_j}<\xi^s_{i_j}\leq q} \Big(c_{1i_j} + c_{1i_j+1} - c_{i_ji_j+1} + F_{i_j+1}(Q)\Big)\, p^s_{i_j} + & \\ \displaystyle\sum_{s:\xi^s_{i_j}\leq q-\theta_{i_j}} F_{i_j+1}(q - \xi^s_{i_j}) p^s_{i_j} & \text{if } j = 2, \ldots, t \\ 0 & \text{if } j = t+1. \end{cases} \tag{3.9}$$

Given equation (3.9) 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:

$$\mathcal{Q}^{k,1} = F_{i_1}(Q). \tag{3.10}$$

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 (3.10).

3.3.3 Volume Based Recourse Policies for the VRPSD

In Subsection 3.3.2, we presented how the recourse function can be efficiently computed using the recursive equation (3.9). However, to evaluate (3.10) 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 $\vec{v} = (v_1 = v_{i_1}, v_{i_2}, \ldots, v_{i_t}, v_{i_{t+1}} = v_1)$, 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: $\pi_1(\vec{v}) = (\theta_{i_2} = \delta Q, \ldots, \theta_{i_j} = \delta Q, \ldots, \theta_{i_t} = 0)$. 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 $i = 1, \ldots, n$. 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 $\vec{v}$: $\pi_2(\vec{v}) = (\theta_{i_2} = \eta\mathbb{E}(\xi_{i_3}), \ldots, \theta_{i_j} = \eta\mathbb{E}(\xi_{i_{j+1}}), \ldots, \theta_{i_t} = 0)$. 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 3.5.

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 $\lambda \in \left[0, \dfrac{Q}{\sum_{r=i_{j+1}}^{i_t} \mathbb{E}(\xi_r)}\right]$. 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.

3.4 The Solution Method

To solve model (3.1)-(3.7), 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 [20], [35] and [28]). 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 3.4.1), and the definition of partial routes and the lower bounding functional cuts (Subsection 3.4.2). We then develop lower bounding strategies that enable the application of the LBF cuts for the present problem (Section 3.4.3).

3.4.1 The Integer L-shaped Algorithm

Model (3.1)-(3.7) 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 (3.4)). We recall that the computation of the recourse cost for a given route was discussed in section 3.3.2. To efficiently solve the model, the Integer L-shaped algorithm, which was originally proposed by [34], 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):

$$CP^\nu : \min_{x,\Theta} \quad \sum_{i<j} c_{ij}x_{ij} + \Theta \tag{3.11}$$

subject to (3.2), (3.3), (3.5), (3.6),

$$\sum_{v_i,v_j \in S^k} x_{ij} \leq |S^k| - \left\lceil \frac{\sum_{v_i \in S^k} \mathbb{E}(\xi_i)}{Q} \right\rceil \forall k \in \mathbf{ST}^{\nu-1}, S^k \subset \mathcal{V} \setminus \{v_1\}, 2 \leq |S^k| \leq n-2, \tag{3.12}$$

$$L + (\Theta_p^q - L)\left(\sum_{h \in \mathbf{PR}^q} W_p^h(x) - |\mathbf{PR}^q| + 1 \right) \leq \Theta \quad \forall q \in \mathbf{PS}^{\nu-1}, p \in \{\alpha, \beta, \gamma\}, \tag{3.13}$$

$$L \leq \Theta \tag{3.14}$$

$$\sum_{\substack{1 \leq i \leq j \\ x_{ij}^f = 1}} x_{ij} \leq \sum_{1 \leq i \leq j} x_{ij}^f - 1 \qquad \forall f \in \mathbf{OC}^{\nu-1}. \tag{3.15}$$

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 (3.2), (3.3), (3.5) and (3.6). 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 (3.12). 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 (3.12).

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 (3.13), and a general lower bound L that is valid over all feasible first-stage solutions, constraint (3.14). As will be detailed in Sections 3.4.2, 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 3.4.2). 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 (3.15) 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 (3.15) 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 [38] 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 [28] 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 [28] for further details).

3.4.2 Lower Bounding Functionals

The LBFs (3.13) are generated based on general partial routes. These were initially proposed by Hjorring and Holt [27] 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. [35]. The concept of partial routes was further elaborated by Jabali et al. [28], 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 [27]. In this paper, we employ the LBFs proposed by Jabali et al. [28]. In what follows, we define the LBFs using the notation proposed by Jabali et al. [28], 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 (3.2), 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. [28] is used in this paper to detect such partial routes. For $h \in \mathbf{PR}^\nu$, let κ denote the number of chains and $\kappa - 1$ 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,

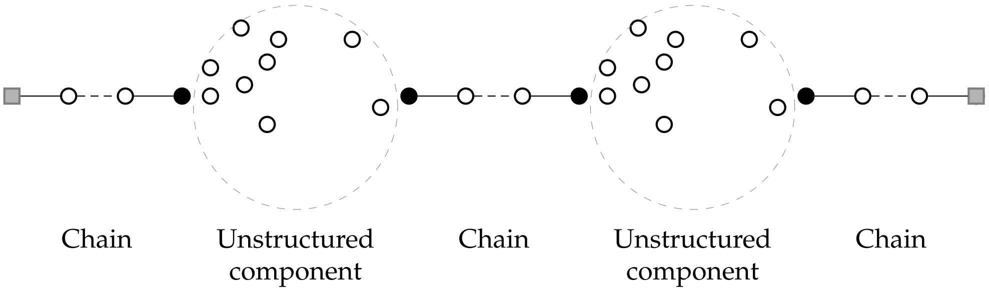


Figure 3.2 – A general partial route h composed of sequenced and unsequenced sets.

$$\sum_{(v_i,v_j)\in S^t_h} x^\nu_{ij} = |S^t_h| - 1, \ \forall t = 1, \dots, \kappa. \tag{3.16}$$

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

$$\sum_{v_i, v_j \in U_h^t} x_{ij}^{\nu} = |U_h^t| - 1,\ \forall t = 1, \ldots, \kappa - 1. \tag{3.17}$$

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

$$\sum_{v_j \in U_h^t} x_{h_l^t, j}^{\nu} = 1,\ \forall t \leq \kappa - 1, \tag{3.18}$$

and

$$\sum_{v_j \in U_h^{t-1}} x_{h_1^t, j}^{\nu} = 1,\ \forall t \geq 2 \tag{3.19}$$

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 (3.3a) 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 [27]. Figure (3.3b) illustrates the case of a β-route topology, where the actual alternation of CVSs and UVSs is maintained. Figure (3.3c) 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 (3.20), and recall its purpose in the LBF cuts, i.e., constraints (3.13). Finally, in Section 3.4.3 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.

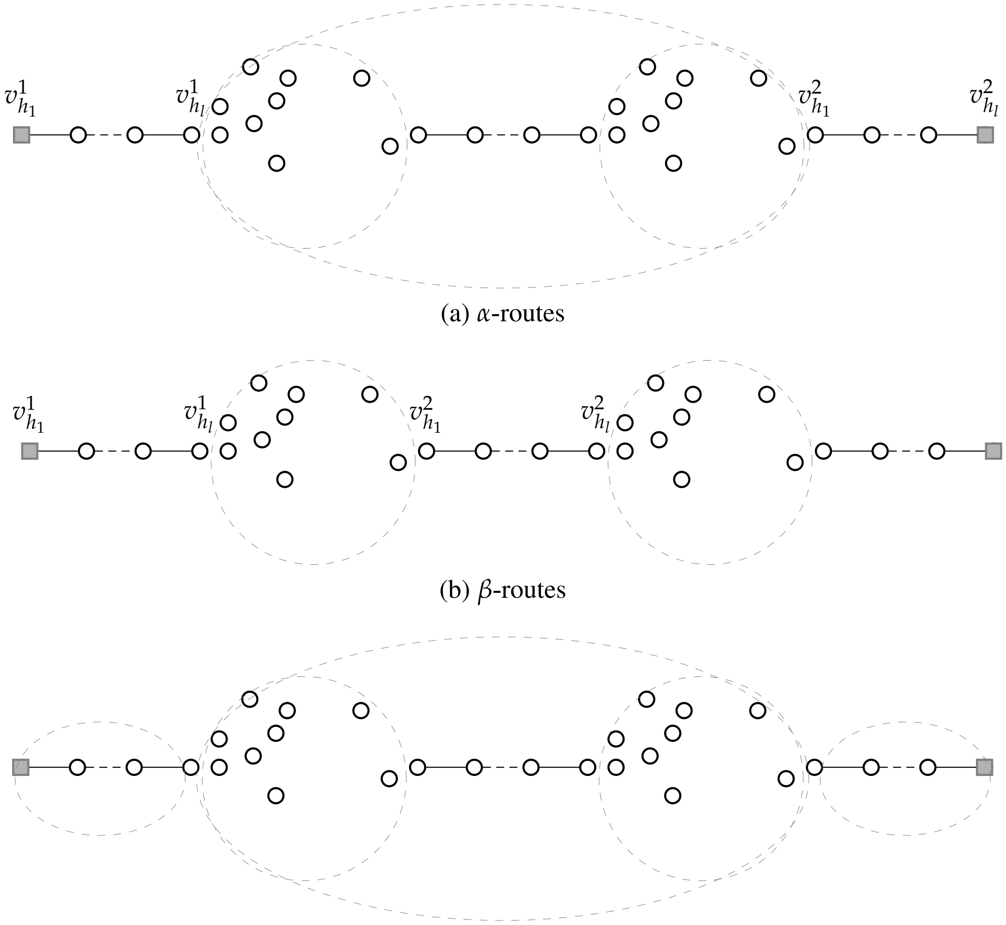
Given a general partial route h, the choice of a topology $p \in \{\alpha, \beta, \gamma\}$ defines the specific succession of CVSs and UVSs that are used to produce the LBF cut. Specifically, a topology fixes the vertices that are included in sets S_h^t, for $t = 1, \ldots, \kappa$, and U_h^t, for $t = 1, \ldots, \kappa - 1$. The functional $W_p^h(x)$, introduced by Jabali et al. [28], is defined as follows,

$$
\begin{aligned}
W_p^h(x) = {} & \sum_{t=1}^{\kappa} \sum_{\substack{(v_i,v_j)\in S_h^t \\ v_i \neq v_1}} 3x_{ij} + \sum_{(v_1,v_j)\in S_h^1} x_{1j} + \sum_{(v_1,v_j)\in S_h^{\kappa}} x_{1j} + \sum_{t=1}^{\kappa-1} \sum_{v_i,v_j\in U_h^t} 3x_{ij} \\
& + \sum_{t=1}^{\kappa-1} \sum_{\substack{v_j\in U_h^t \\ v_{h_l}^t \neq v_1}} 3x_{h_l^t j} + \sum_{t=2}^{\kappa} \sum_{\substack{v_j\in U_h^{t-1} \\ v_{h_1}^t \neq v_1}} 3x_{h_1^t j} + \sum_{\substack{v_j\in U_h^1 \\ v_{h_l}^1 = v_1}} x_{h_l^1 j} + \sum_{\substack{v_j\in U_h^{b-1} \\ v_{h_1}^{\kappa} = v_1 \\ v_{h_1}^{\kappa-1} \neq v_1}} x_{h_1^{\kappa} j} \\
& - (3|R_h| - 5).
\end{aligned}
\tag{3.20}
$$

We refer the reader to Jabali et al. [28] for the proof of validity of equation (3.20) as a component of the LBF cut (3.13). 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, $\sum\limits_{h\in\mathbf{PR}^q} W_p^h(x) = |\mathbf{PR}^q|$ 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 $\Theta_p^q \leq \Theta$.

3.4.3 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 when topology $p \in \{\alpha, \beta, \gamma\}$ is applied to generate an LBF cut (3.13). Moreover, the bound Θ_p^q, which is included in (3.13), is fixed to the sum of the lower bounds associated with the different partial routes associated with q, i.e., $\Theta_p^q = \sum\limits_{h\in\mathbf{PR}^q} \Theta_p^{qh}$. In the following, to alleviate the notation, we will drop the index q and simply refer to the lower bound Θ_p^h (i.e., a partial route is always associated with a partial solution). Furthermore, we focus on deriving value Θ_α^h (i.e., the specific topology $p = \alpha$). This is motivated by the fact that the computation of Θ_α^h can be easily generalized to evaluate both Θ_β^h and Θ_γ^h, considering that topologies β and γ can be viewed as containing successive α-route



(a) α-routes

(b) β-routes

(c) γ-routes

Figure 3.3 – Partial route topologies.

structures. We next present the strategy to compute Θ_α^h 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_α^h 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 $v^1_{|S^1_h|}$ and $v^2_{|S^2_h|}$ 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,

$$\tilde{h} = (v_1 = v_{i_1}, \ldots, v_{i_{j-l}}, \square_{i_{j-l+1}}, \square_{i_{j-l+2}}, \ldots, \square_{i_j}, v_{i_{j+1}}, \ldots, v_{i_{t+1}} = v_1), \qquad (3.21)$$

where each possible ordering of the l unsequenced customers included in U^1_h 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 (3.9), 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 (3.9) in a manner that yields a bound on the unsequenced customers in U^1_h. 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 $v_{u_e} \in U^1_h$ 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 3.4.1 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 3.4.1 to bound artificial route $\tilde{h}$. This is done in two main steps, in Lemma 3.4.2 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 3.4.3 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 3.4.1. *Let q denote the residual capacity of the vehicle upon arriving at v_{i_k}. Let*

$$\hat{F}_{i_k}(q) = \begin{cases} \widetilde{F}_{i_{k+1}}(q) & \textit{if } k = 1 \\ \displaystyle\sum_{s:\xi^s_{i_k} > q} \left(b + 2c_{1i_k} + \widetilde{F}_{i_{k+1}}(Q + q - \xi^s_{i_k})\right) p^s_{i_k} + & \\ \displaystyle\sum_{s:q-\underline{\theta}_{\vec{v}} < \xi^s_{i_k} \leq q} \left(\tilde{c}_{i_k} + \widetilde{F}_{i_{k+1}}(Q)\right) p^s_{i_k} + & \\ \displaystyle\sum_{s:\xi^s_{i_k} \leq q - \overline{\theta}_{\vec{v}}} \widetilde{F}_{i_{k+1}}(q - \xi^s_{i_k}) p^s_{i_k} & \textit{if } k = 2, \ldots, t \\ 0 & \textit{if } k = t+1, \end{cases} \tag{3.22}$$

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 (3.9) as

$$F_{i_k}(q) = \begin{cases} F_{i_k+1}(q) & \text{if } k = 1 \\ \displaystyle\sum_{s:\xi^s_{i_k} > q} \left(b + 2c_{1i_k} + F_{i_{k+1}}(Q + q - \xi^s_{i_k})\right) p^s_{i_k} + & \\ \displaystyle\sum_{s:q-\theta_{i_k} < \xi^s_{i_k} \leq q} \left(c_{1i_k} + c_{1i_{k+1}} - c_{i_k i_{k+1}} + F_{i_{k+1}}(Q)\right) p^s_{i_k} + & \\ \displaystyle\sum_{s:\xi^s_{i_k} \leq q - \theta_{i_k}} F_{i_{k+1}}(q - \xi^s_{i_k}) p^s_{i_k} & \text{if } k = 2, \ldots, t \\ 0 & \text{if } k = t+1. \end{cases}$$

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 (3.9) (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 3.4.2. *A lower bound on the expected recourse cost for the j^{th} position in the artificial route $\tilde{h}$ can be defined as follows:*

$$\widetilde{F}_{i_j}(q) = \min_{v_{u_e} \in U_h^1} F_{i_j}(q)|_{i_j := u_e} \, \forall q \tag{3.23}$$

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 (3.9).

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 3.4.2 within Lemma 3.4.1, a valid lower bound can be derived for the positions not yet sequenced in $\tilde{h}$, i.e., $(\square_{i_{j-l+1}}, \square_{i_{j-l+2}}, \ldots, \square_{i_{j-1}})$. Therefore, at the $j-1^{\text{th}}$ position, Lemma 3.4.2 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 3.4.3. *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:*

$$\widetilde{F}_{i_k}(q) = \min_{v_{u_e} \in U_h^1} \hat{F}_{i_k}(q)|_{i_k := u_e} \, \forall q, \tag{3.24}$$

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

$$\hat{F}_{i_k}(q)|_{i_k:=u_e} = \begin{cases} \displaystyle\sum_{s:\xi^s_{u_e}>q} \left(b + 2c_{1u_e} + \widetilde{F}_{i_{k+1}}(Q+q-\xi^s_{u_e})\right) p^s_{u_e} + \\ \displaystyle\sum_{s:q-\underline{\theta}_{U^1_h}<\xi^s_{u_e}\leq q} \left(\tilde{c}_{u_e} + \widetilde{F}_{i_{k+1}}(Q)\right) p^s_{u_e} + \\ \displaystyle\sum_{s:\xi^s_{u_e}\leq q-\overline{\theta}_{U^1_h}} \widetilde{F}_{i_{k+1}}(q-\xi^s_{i_k}) p^s_{u_e} \end{cases} \tag{3.25}$$

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

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

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

Proof. Let us consider position i_{j-1}, where the valid lower bound $\widetilde{F}_{i_j}(.)$ is assumed known, considering Lemma 3.4.2. Let

$$\hat{F}_{i_{j-1}}(q)|_{i_{j-1}:=u_e} = \begin{cases} \displaystyle\sum_{s:\xi^s_{u_e}>q} \left(b + 2c_{1u_e} + \widetilde{F}_{i_j}(Q+q-\xi^s_{u_e})\right) p^s_{u_e} + \\ \displaystyle\sum_{s:q-\underline{\theta}_{U^1_h}<\xi^s_{u_e}\leq q} \left(\tilde{c}_{u_e} + \widetilde{F}_{i_j}(Q)\right) p^s_{u_e} + \\ \displaystyle\sum_{s:\xi^s_{u_e}\leq q-\overline{\theta}_{U^1_h}} \widetilde{F}_{i_j}(q-\xi^s_{i_k}) p^s_{u_e} \end{cases}$$

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 3.4.1). By defining $\widetilde{F}_{i_{j-1}}(q) = \min_{v_{u_e}\in U^1_h} \hat{F}_{i_{j-1}}(q)|_{i_{j-1}:=u_e}$, 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^1_h), 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

$$\hat{F}_{i_{j-l}}(q) = \begin{cases} \displaystyle\sum_{s:\xi^s_{i_{j-l}}>q} \left(b + 2c_{1i_{j-l}} + \widetilde{F}_{i_{j-l+1}}(Q + q - \xi^s_{i_{j-l}})\right) p^s_{i_{j-l}} + \\ \displaystyle\sum_{s:q-\underline{\theta}_{U^1_h}<\xi^s_{i_{j-l}}\leq q} \left(\tilde{c}_{i_{j-l}} + \widetilde{F}_{i_{j-l+1}}(Q)\right) p^s_{i_{j-l}} + \\ \displaystyle\sum_{s:\xi^s_{i_{j-l}}\leq q-\overline{\theta}_{U^1_h}} \widetilde{F}_{i_{j-l+1}}(q - \xi^s_{i_k}) p^s_{i_{j-l}} \end{cases} \tag{3.26}$$

given that $\widetilde{F}_{i_{j-l+1}}(q)$ for all q is computed using Lemma 3.4.3 and where $\tilde{c}_{i_{j-l}} = \min\limits_{v_{u_e}\in U^1_h} \{c_{1,i_{j-l}} + c_{1,u_e} - c_{i_{j-l},u_e}\}$ defines the minimum PR trip cost that could be incurred from $v_{i_{j-l}}$ into U^1_h.

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 (3.9) 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 $\widetilde{F}_{i_1}(Q) = \tilde{\mathcal{Q}}^{k,1}_{\tilde{h}}$, can be used to complete the computation of the lower bound value. As for obtaining value $\tilde{\mathcal{Q}}^{k,2}_{\tilde{h}}$, we simply reverse the artificial route and apply the same computation. Therefore, $\Theta^h_\alpha = \min\{\tilde{\mathcal{Q}}^{k,1}_{\tilde{h}}, \tilde{\mathcal{Q}}^{k,2}_{\tilde{h}}\}$ 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., $\underline{\theta}_{U^1_h}$ and $\overline{\theta}_{U^1_h}$ in Lemma 3.4.3). These threshold values now need to be determined according to the specific positions associated with U^1_h. Let us define $\underline{\theta}^k_{U^1_h}$ and $\overline{\theta}^k_{U^1_h}$ as the aforementioned threshold values associated with position k, for $j-l+1 \leq k < j$. To express these values, we define 1st, 2nd, …, $l-1$th minimum and maximum expected demands associated with the customers included in U^1_h 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

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

Finally, to compute $\hat{F}_{i_{j-l}}(q)$ using (3.26), under policy π_3, values $\theta_{U_h^1}^{min}$ and $\theta_{U_h^1}^{max}$ are simply set to

$$\theta_{U_h^1}^{min} = \theta_{U_h^1}^{max} = \lambda\big(\sum_{v_{u_e} \in U_h^1} \mathbb{E}(\xi_{v_{u_e}}) + \sum_{r=i_{j+1}}^{i_t} \mathbb{E}(\xi_r)\big).$$

We have presented the computation of the bounds associated with Θ_α^h. This computation is generalized, to both Θ_β^h and Θ_γ^h, as these can be viewed as successive α-route structures.

3.5 Numerical Results

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 (3.I). 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 §3.3.3, 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 (3.4) are generated using the CVRPSEP package of [38] and the branching procedure, which is used for the *L*-shaped algorithm, is implemented using the OOBB package developed by [24]. 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.

The obtained results are analyzed in the next two subsections. In Subsection 3.5.1, 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 3.5.2, a solution cost assessment is conducted for the proposed policies.

3.5.1 Computational Policy Analysis

The results obtained for all numerical experiments are summarized in Tables 3.II, 3.III, and 3.IV, 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 3.II, 3.III, and 3.IV, one first observes the general trend that was previously reported by Gendreau et al. [20], Laporte et al. [35], and Jabali et al. [28] 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 3.II, 3.III, and 3.IV, 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

Table 3.I – 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	$2, 3, 4$	$0.90, 0.92, 0.94, 0.96$

Table 3.II – 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.III – 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 3.IV – 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		

and 593 runs using π_3. From these results, it clearly appears that the Integer L-shaped algorithm is most efficient 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.

3.5.2 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. [5]. 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 3.V, 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 3.V, 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 3.IV, 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.

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 3.VI, 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.

Finally, when comparing the proposed policies to the optimal restocking one, it can

Table 3.V – The ratio $\frac{EBF_c}{EBF_r}$

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

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.

3.6 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

Table 3.VI – 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%

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.

CHAPTER 4

ARTICLE 2: A HYBRID RECOURSE POLICY FOR THE VEHICLE ROUTING PROBLEM WITH STOCHASTIC DEMANDS

Chapter notes: This chapter has been submitted to the *EURO Journal on Transportation and Logistics*. Preliminary work was presented at the following conferences:

- Optimization Days 2016, Montreal, Canada, May 2-4, 2016
- APORS 2015, Kuching Sarawak, Malaysia, August 1-3, 2015
- EURO 2015, Glasgow, Scotland, July 12-15, 2015
- CORS/INFORMS 2015, Montreal, Canada, June 14 - 17, 2015

Contribution:

- The general orientations of paper are proposed by the supervisors.
- Conducting the research including the modelling of ideas, implementations and coding parts is carried out by student. The draft versions of the three papers are written by student and then are modified by supervisors.

A Hybrid Recourse Policy for the Vehicle Routing Problem with Stochastic Demands

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

[1]Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport (CIRRELT), C.P. 6128, succ. Centre-ville, Montréal, Québec H3C 3J7, Canada

[2]Département d'informatique et de recherche operationnelle, Université de Montréal, C.P. 6128, succ. Centre-ville, Montréal, Québec H3C 3J7, Canada

[3]Département de mathématiques et de génie industriel, Polytechnique de Montréal, C.P. 6079, succ. Centre-ville, Montréal, Québec H3C 3J7, Canada

[4]Dipartimento di elettronica, informazione e bioingegneria, Politecnico di Milano, Piazza Leonardo da Vinci, 32, Milano 20133, Italy

[5]Département de management et technologie, École des sciences de la gestion, Université du Québec à Montréal, C.P. 8888, succ. Centre-ville, Montréal, Québec H3C 3P8, Canada

Abstract

In this paper we propose a new recourse policy for the vehicle routing problem with stochastic demands (VRPSD). In this routing problem customer demands are characterized by known probability distributions. The objective of the problem is to plan routes minimizing the travel cost and the expect recourse cost. The latter cost is a result of a predetermined recourse policy designed to handle route failures. In the relevant literature there are three types of recourse policies i) classical, where stock outs at customers are handled by return trips to the depot ii) optimal restocking, where preventive restocking trips to the depot are performed based on optimized customer-specific thresholds, and stock outs are handled by return trips to the depot iii) rule-based policies, where preventive restocking trips are performed based on thresholds established by preset rules, and stock outs are handled by performing return trips to the depot. The latter policy enables a company to define its recourse policy based on its operational conventions. We first propose a taxonomy that groups rules-based policies into three classes. We then propose the first hybrid recourse policy, which simultaneously combines two of these classes, namely risk and distance. We propose an exact solution algorithm for the VRPSD with this hybrid recourse policy. We conduct a broad range of computational experiments. For certain experimental configurations, the exact algorithm solves to optimality up to 79 percent of the instances. Furthermore, the algorithm is able to solve instances with up to 60 customers. Compared to the classical recourse policy, on average, our hybrid policy results in a lower number of expected failures. Finally, we show that when the optimal routes of the hybrid policy are operated under the classical policy they produce higher expected recourse costs on average. However, operating the same routes under the optimal restocking policy yields an average marginal cost difference with respect to our hybrid policy.

Keywords: Hybrid recourse policy; Preventive restocking; Operational rules; Vehicle routing problem with stochastic demands; Partial routes; L-shaped algorithm; Lower bounding functionals

4.1 Introduction.

The extensively-studied vehicle routing problem (VRP) aims to route a set of homogeneous vehicles with limited capacity to serve the demand of a set of customers. The objective of the VRP is to minimize the total distance driven by the vehicles such that each vehicle starts and ends its route at a given depot, each customer must be visited once by a single vehicle, and the total demand of a route does not exceed the vehicle capacity. In an attempt to capture more realistic features, a number of variants of the VRP have been proposed (see Toth and Vigo [55] for an extensive review). One particular drawback of the VRP lies in the assumption that all problem parameters are deterministic. In reality, several parameters such as customer demands or travel time are stochastic. Modelling the VRP while using deterministic approximation of stochastic parameters, e.g., using the mean value as an approximation, may result in arbitrarily bad-quality solutions (Louveaux [36]). Therefore, an ever growing class of problems, referred to as stochastic vehicle routing problem (SVRP), has been receiving increasing attention (Gendreau et al. [23]). Modelling stochasticity in practice implies that a sufficient amount of data is gathered to describe the probability distribution of uncertain parameters. The ever growing availability of data enables practitioners to construct and validate such probability distributions, thus the study of SVRP is rather timely. While different modelling paradigms exist for handling the SVRP, their guiding principle is to capitalize upon the knowledge of the distribution functions that define stochastic parameters in order to produce solutions that are more suitable for the stochastic environment.

In this paper we study the vehicle routing with stochastic demands (VRPSD), in which the demand of each customer follows a customer-specific probability distribution. Moreover, we assume that the precise demand value of a customer is only revealed when it is first visited by a vehicle. The VRPSD can be observed in a number of realistic applications, such as in home oil delivery (Chepuri and Homem-De-Mello [12]), garbage collection (Yang et al. [57]) and the collection of money from banks (Lambert et al. [31]).

Several modelling paradigms have been proposed for the VRPSD, see Gendreau et al. [22] for an extensive review. In this paper we use the *a priori* modelling paradigm, which was originally put forward by Bertsimas et al. [4]. In the context of VRPSD, the a priori paradigm decomposes the problem into two stages. The first-stage consists of determining a set of planned a priori vehicle routes, without the knowledge of the precise demand values of the customers. These values are revealed in the second-stage when routes are performed. Due to the stochastic nature of the demands, an a priori route may fail at a specific customer if its revealed demand exceeds the residual vehicle capacity, i.e., the remaining capacity of the vehicle upon arriving to the customer location. In such cases, a route failure happens (Dror and Trudeau [16]) and is handled by recourse actions stemming from a recourse policy.

Two main recourse actions for the VRPSD are found in the literature. In the first, one can recover routing feasibility through the use of a reactive replenishment trip to the depot after a failure is observed. Namely, in the case that the residual capacity is less than the observed customer demand, the vehicle performs a *back-and-forth* (BF) trip to the depot, where the vehicle is replenished and returns to the customer location where the failure occurred, and if possible continues visiting customers in the order of the planned route. In the case that the residual capacity is precisely equal to the observed customer demand, and this customer is not the last customer on the planned route, the vehicle performs a *restocking trip* (RT) to the depot and then proceeds to unvisited customers in the order of the planned route, see Gendreau et al. [20], Hjorring and Holt [27]. In the second type of recourse action, one anticipates route failures and may execute a proactive replenishment trip to the depot before an actual route failure occurs. In this case, the vehicle executes a *preventive restocking* (PR) trip, i.e., returns to the depot with residual capacity and once replenished continues visiting customers in the order of the planned route. PR helps in avoiding costly failures as shown by Yee and Golden [58] and Yang et al. [57]. Both these recourse actions operate on each route independently, implying that a vehicle designated to serving a route in the first-stage is exclusively serving the customers included in the route during the second-stage. Thus, these recourse actions preserve person-oriented consistency, which entails that customers are served by a specific driver whenever service is required ([30]).

The a priori formulation for the VRPSD works with a predetermined recourse policy, which dictates when recourse actions are performed. There are three types of recourse policies used in this context. The *classical recourse*, according to which a route failure or an exact stock out trigger a BF or RT (when needed), respectively. This purely reactive policy is the most studied version of the VRPSD (Gendreau et al. [22]). Several exact algorithms have been proposed for the VRPSD with the classical recourse. Gendreau et al. [20], Laporte et al. [35], and Jabali et al. [28] use the *L*-shaped algorithm while, Christiansen and Lysgaard [13] and Gauvin et al. [19] use column generation approaches. Heuristic algorithms were also proposed for this problem, e.g., Gendreau et al. [21], Rei et al. [44], and [40].

The second type of recourse policy is the *optimal restocking* policy, which employs PR and BF actions. Given a planned route, this policy computes optimal customer-specific thresholds based on which a vehicle performs PR trips. Specifically, when the residual capacity is less than the customer's threshold but greater or equal to the customer's demand, a PR trip is performed. In the case that the customer's demand exceeds vehicle residual capacity a BF trip is performed. The optimal restocking policy was first proposed by Yee and Golden [58]. Several heuristic algorithms are proposed for this policy. A cyclic heuristic (Bertsimas et al. [5]), a local search heuristic (Yang et al. [57]), and a metaheuristic (Bianchi et al. [7]).

The third recourse policy is the *rule-based recourse* policy, which was recently coined by Salavati-Khoshghalb et al. [45]. Similar to the optimal restocking policy, PR and BF actions are performed. However, the former is governed by a family of restocking rules based on volume related measures. Within this family, three rule-based restocking policies are introduced: residual vehicle capacity, expected demand of the next customer, and expected demands of unvisited customers. These policies operate with preset rules that determine the customer thresholds for performing PR trips. For example, the first rule-based restocking policy requires a PR trip to be preformed whenever the residual capacity of the vehicle falls below a certain percentage of its total capacity. An exact algorithm capable of handling the three rule-based policies was developed.

It is worth noting that more intricate recourse policies such as route reoptimization ([49]) have been proposed in the literature. From a cost perspective, reoptimizing routing decisions as stochastic information is revealed is a better theoretical alternative to the three previously discussed policies. However, solving the VRPSD with reoptimization is challenging. The heuristic described in Secomandi and Margot [49] has been implemented for the single vehicle case only. Moreover, reoptimizing routing implies that customers are not served by the same drivers consistently, the actual arrival time at a customer location may be very variable. To this end, we argue that the a priori paradigm fits practical contexts where one seeks to design a tactical set of fixed routes, which are minimally altered on a daily basis. Such tactical routes are suitable when preserving consistency in routing operations is desired (see Salavati-Khoshghalb et al. [45] for further motivation).

Transportation companies often use operational conventions when dealing with uncertainty. Rule-based policies facilitate in reflecting such conventions in a routing environment, which is not necessarily the case in the optimal restocking policy (see Salavati-Khoshghalb et al. [45] for a general motivation for rule-based policies). Furthermore, rule-based policies allow companies to control the risk of encountering failures, and thus better tailor recourse actions to customer service conventions.

We first propose a taxonomy that groups rule-based policies into three classes. We then introduce a *hybrid recourse policy*, which combines rules from two of these classes. In particular, this hybrid policy triggers replenishment decisions based on risk and distance measures. For a given route, the risk measure computes the risk of failure at the next customer. This is compared with predetermined thresholds corresponding to a *minimum restocking threshold* and a *maximum proceeding threshold*. If the risk of failure is greater than the former threshold, then the vehicle executes a PR trip, and if the risk of failure is less than the latter threshold, then the vehicle proceeds with the planned route. In all other cases, (i.e., where the risk of failure is between the maximum proceeding threshold and the minimum restocking threshold) we employ a distance measure, which compares the cost of a PR trip at the current customer with the average cost of future failures resulting from BF trips. For simplicity, in what follows we refer to the hybrid risk-and-distance policy as the hybrid policy. We develop an exact algorithm to solve the VRPSD with the hybrid recourse policy. Furthermore, extensive numerical experiments are performed, in which we demonstrate the effectiveness of the solution algorithm and compare the hybrid recourse policy with other recourse policies.

The remainder of this paper is organized as follows. In Section §4.2 we present the VRPSD model, provide a taxonomy for rule-based recourse policies, and present our hybrid recourse policy. We elaborate the exact solution algorithm in Section §4.3. Numerical experiments are presented in Section §4.4. Finally, we present our conclusions and future research directions in Section §4.5.

4.2 The vehicle routing problem with stochastic demands and a hybrid recourse policy

In section §4.2.1, we present the two-stage stochastic programming formulation for the VRPSD, initially proposed by Laporte et al. [35]. We then present a concise taxonomy for the rule-based policies in Section §4.2.2. Based on this taxonomy we elaborate the proposed hybrid recourse policy in Section §4.2.3.

4.2.1 The a priori model for the VRPSD

In this section we present the a priori model for the VRPSD using the original notation defined by Laporte et al. [35]. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a complete undirected graph, where $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$ is the set of vertices and $\mathcal{E} = \{(v_i, v_j) | v_i, v_j \in \mathcal{V}, i < j\}$ is the edge set. The cost of travelling along edge (v_i, v_j) is denoted by c_{ij}. The depot is denoted by v_1 and the set of customers is $\mathcal{V} \setminus \{v_1\}$. There are m vehicles at the depot, each of which has a capacity of Q. The demand of a customer v_i is ξ_i and is assumed to follow a discrete probability distribution with a finite support defined as $\{\xi_i^1, \xi_i^2, \dots, \xi_i^{s^i}\}$, where values are indicated by increasing order, $\xi_i^1 > 0$ and $\xi_i^{s^i} < Q$. Let p_i^l denote the probability that the realized demand at customer v_i is ξ_i^l.

The decision variable x_{ij} $(i < j)$ is an integer equal to the number of times edge (v_i, v_j) appears in the first-stage solution, i.e., x_{ij} must be interpreted as x_{ji} for $i > j$. The variable x_{1j} may take the values $\{0, 1, 2\}$, where $x_{1j} = 2$ expresses a route visiting a single customer. The variable x_{ij} is binary when $i, j > 1$. As in Laporte et al. [35]

and Jabali et al. [28], we assume that the expected demand of an a priori route does not exceed the vehicle capacity. This assumption forbids the generation of routes that are likely to systematically fail. Furthermore, let $\mathcal{Q}(x)$ denote the expected second stage cost of solution x. The a priori model for the VRPSD can be formulated as follows:

$$
\begin{aligned}
&\underset{x}{\text{minimize}} && \sum_{i<j} c_{ij}x_{ij} + \mathcal{Q}(x) && && (4.1)\\
&\text{subject to} && \sum_{j=2}^{n} x_{1j} = 2m, && && (4.2)\\
& && \sum_{i<k} x_{ik} + \sum_{k<j} x_{kj} = 2, && k = 2,\dots,n && (4.3)\\
& && \sum_{v_i,v_j \in S} x_{ij} \leq |S| - \left\lceil \frac{\sum_{v_i \in S} \mathbb{E}(\xi_i)}{Q} \right\rceil, && (S \subset \mathcal{V} \setminus \{v_1\}; 2 \leq |S| \leq n-2) && \\
& && && && (4.4)\\
& && 0 \leq x_{ij} \leq 1, && 2 \leq i < j < n && (4.5)\\
& && 0 \leq x_{1j} \leq 2, && j = 2,\dots,n && (4.6)\\
& && x = (x_{ij}), && \text{integer} && (4.7)
\end{aligned}
$$

The objective function (4.1) consists of minimizing the first-stage cost and the second-stage cost. The former is the cost of the a priori routes, while the latter is their associated recourse cost. Constraints (4.2) and (4.3) establish the degree of the vertices. Constraints (4.4) eliminate subtours, and ensure that the total expected demand of each route is less or equal to Q. Finally, constraints (4.5), (4.6) and (4.7) define the domains of the decision variables.

Given that the considered recourse actions are performed independently by the vehicle performing the a priori route, $\mathcal{Q}(x)$ is separable with respect to the routes. 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 determined. Let $\mathcal{Q}^{r,\delta}$ be the expected recourse cost of the r^{th} vehicle-route when performed in orientation δ ($\delta = 1, 2$). Thus,

$$\mathcal{Q}(x) = \sum_{r=1}^{m} \min\{\mathcal{Q}^{r,1}, \mathcal{Q}^{r,2}\}. \tag{4.8}$$

The computation of $\mathcal{Q}^{r,\delta}$ is elaborated in section §4.2.3.

4.2.2 A taxonomy for rule based policies

The use of rule-based policies in VRPSD implies that recourse actions are taken based on a set of preset rules. These rules establish customer specific thresholds that govern when a PR trip is executed. 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. To do so, we present a concise taxonomy for the considered policies and then clearly define the hybrid policy considered in the present paper.

We propose a taxonomy that groups the possible policies in three general classes: (i) *volume-based policies*, (ii) *risk-based policies* and (iii) *distance-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. Alternatively, risk-based policies derive the thresholds on the basis of the probability of failure at the next or at the following customers along the considered route. In this case, a company can use the available knowledge regarding the distributions of the demands of its customers to evaluate the risk of observing failures when performing a route. Risk-based policies can then apply operational rules that express varying levels of risk aversion with regards to route failures. Such rules would call for a PR trip to be performed whenever the probability of failure exceeds a predetermined level. Distance-based policies consider the distance between the customers and the depot to obtain the thresholds. The general principle being applied here is that it is preferable to carry out a PR trip from a customer located close to the depot than to risk a failure at a more distant one. Finally, *hybrid policies* can also be defined by combining the previous ones.

In this paper, we employ a hybrid risk-and-distance-based policy to govern recourse actions. Therefore, we propose the first hybrid recourse policy that combines two classes of policies. Our policy uses post-realization information, i.e., the residual capacity after serving a customer, to determine recourse actions which, in return are used to compute the expected recourse cost. In what follows, we present our hybrid rule-based recourse policy and the exact computation of its expected recourse cost.

4.2.3 A Hybrid Recourse Policy for the VRPSD

Given a route one can measure the risk of route failure at the next customer. In this context, we identify three categories of action. If the risk is too high, the vehicle executes a PR trip, and respectively if the failure risk is too low, then the vehicle proceeds to the unvisited customers. For intermediate cases, we combine the defined risk measure with a distance-based measure, according to which a PR trip is performed if deemed beneficial.

We now formulate the risk and distance based measures. We recall that the recourse cost $\mathcal{Q}(x)$ is computed independently for each given route. Given an a priori route $r = (v_1 = v_{r_1}, v_{r_2}, \dots, v_{r_{l-1}}, v_{r_l} = v_1)$, let the vehicle residual capacity upon arrival at the j^{th} customer be q and let ξ_{r_j} be the observed demand. The post realization residual capacity is $\tilde{q} = q - \xi_{r_j}$, given that ξ_{r_j} follows a discrete probability distribution, two cases may occur $\tilde{q} = q - \xi_{r_j} \leq 0$, or $\tilde{q} \geq 1$. If $v_{r_{j+1}} \neq v_1$ and $\tilde{q} = 0$, a RT trip is performed, where the vehicle replenishes at the depot and goes to $v_{r_{j+1}}$. When $\tilde{q} < 0$ the vehicle performs a BF trip to the j^{th}. In this situation, the service of the customer is split, and the overhead of the unloading process is duplicated causing delays and disruptions at the customer location. Therefore, similar to Yang et al. [57], we attribute a penalty cost b to a BF trip. For the case where $\tilde{q} \geq 1$, a decision pertaining to whether a PR trip should be performed, or not, is taken. To take this decision, we defined a risk measure, which is the probability of failure at the subsequent customer and is computed as follows,

$$\mathbb{P}[\xi_{r_{j+1}} > \tilde{q}] = \sum_{l:\xi^l_{r_{j+1}} > \tilde{q}} p^l_{r_{j+1}} \tag{4.9}$$

where, the right-hand-side of equation (4.9) computes the total probability of failure events at the next customer $v_{r_{j+1}}$.

Recourse actions are taken based on a comparison of the resulting risk measure in equation (4.9) with thresholds $\underline{\theta}$ and $\bar{\theta}$. Where $\underline{\theta}$ is the maximum proceeding threshold, and $\bar{\theta}$ is the minimum restocking threshold. If $\mathbb{P}[\xi_{r_{j+1}} > \tilde{q}] \leq \underline{\theta}$ we proceed with the planned route, and if $v_{r_{j+1}} \neq v_1$ and $\mathbb{P}[\xi_{r_{j+1}} > \tilde{q}] \geq \bar{\theta}$ we perform a PR trip. The former case corresponds to having high residual capacity, thus yielding low probability of failure at the next customer, whereas the latter corresponds to the situation of low residual capacity thus yielding high probability of failure at the next customer. If $\underline{\theta} < \mathbb{P}[\xi_{r_j+1} > \tilde{q}] < \bar{\theta}$ the risk of failure is neither too low nor too high. In this case, we employ a distance-based measure in order to determine whether to perform a PR trip. The distance-based measure is based on the expected failure cost at all subsequent customers in the route. Let u_{r_j} be the set of subsequent customers to the j^{th} customer in route r, i.e., $u_{r_j} = \{v_{r_j+1}, \ldots, v_{r_{l-1}}\}$. The distance-based measure is defined as $p^*_{r_j}(\tilde{q})(2\bar{c}_{r_j} + b)$, and is computed as follows,

$$\bar{c}_{r_j} = \frac{\sum\limits_{k \in u_{r_j}} c_{1k}}{|u_{r_j}|}$$

and

$$p^*_{r_j}(\tilde{q}) = \mathbb{P}[\sum_{k \in u_{r_j}} \xi_k > \tilde{q}].$$

The value $2\bar{c}_{r_j} + b$ is the average failure cost incurred by unvisited customers in u_{r_j}, and $p^*_{r_j}(\tilde{q})$ is the probability of failure, while serving customers in u_{r_j} with $\tilde{q}$ units of the residual capacity.

Given the residual capacity $\tilde{q}$ at the j^{th} customer in route r, we introduce the Boolean variable $\text{DP}_{r_j}(\tilde{q})$ as follows,

$$\text{DP}_{r_j}(\tilde{q}) := \begin{cases} \texttt{True} & \text{if } c_{1r_j} + c_{1r_j+1} < c_{r_j r_j+1} + (2\bar{c}_{r_j} + b)p^*_{r_j}(\tilde{q}) \\ \texttt{False} & \text{otherwise} \end{cases} \tag{4.10}$$

In the case that $\mathrm{DP}_{r_j}(\tilde{q})$ is `True` a PR trip is performed, otherwise the vehicle proceeds to the subsequent customer. Let $Q^R_{r_j}$ denote the set of residual capacities at the j^{th} customer in route r for which a PR trip is performed. Furthermore, let $Q^P_{r_j}$ denote the set of residual capacities at the j^{th} customer in route r for which the vehicle proceeds with the planned route. We now define the hybrid policy, which establishes the decision of whether to perform a PR trip or proceed to with the planned route. The hybrid policy is defined as follows,

$$\begin{aligned} Q^R_{r_j} = \Big\{\tilde{q} \in \{0,1,\ldots,Q\} | \mathbb{P}[\xi_{r_j+1} > \tilde{q}] \geq \bar{\theta}\Big\} \bigcup \\ \Big\{\tilde{q} \in \{0,1,\ldots,Q\} | \underline{\theta} < \mathbb{P}[\xi_{r_j+1} > \tilde{q}] < \bar{\theta} \wedge \mathrm{DP}_{r_j}(\tilde{q})\Big\} \end{aligned} \tag{4.11}$$

and

$$\begin{aligned} Q^P_{r_j} = \Big\{\tilde{q} \in \{1,\ldots,Q\} | \mathbb{P}[\xi_{r_j+1} > \tilde{q}] \leq \underline{\theta}\Big\} \bigcup \\ \Big\{\tilde{q} \in \{1,\ldots,Q\} | \underline{\theta} < \mathbb{P}[\xi_{r_j+1} > \tilde{q}] < \bar{\theta} \wedge \overline{\mathrm{DP}}_{r_j}(\tilde{q})\Big\}. \end{aligned} \tag{4.12}$$

Where $\overline{\mathrm{DP}}_{r_j}(\tilde{q})$ is defined as the complement of $\mathrm{DP}_{r_j}(\tilde{q})$. Therefore, $Q^R_{r_j}$ and $Q^P_{r_j}$ are two mutually exclusive subsets.

The expected recourse cost upon arrival at the j^{th} customer in route r with q units of residual capacity is $F_{r_j}(q)$. $F^{post}_{r_j}(\tilde{q})$ is the recourse cost after the demand realization at r_j. Therefore,

$$F_{r_j}(q) = \mathbb{E}_{\xi_{r_j}}\left[F^{post}_{r_j}(\tilde{q})\right] \ \forall \tilde{q} = q - \xi_{r_j}, \tag{4.13}$$

where $\xi_{r_j} \in \{\xi^1_{r_j}, \xi^2_{r_j}, \ldots, \xi^l_{r_j}, \ldots, \xi^{s^{r_j}}_{r_j}\}$. Following the definition of our hybrid recourse policy, $F^{post}_{r_j}(\tilde{q})$ can be expressed as follows.

$$F_{r_j}^{post}(\tilde{q}) = \begin{cases} b + 2c_{1r_j} + F_{r_{j+1}}(Q + \tilde{q}) & \text{if } \tilde{q} < 0 \quad (4.14a) \\ c_{1r_j} + c_{1r_j+1} - c_{r_jr_j+1} + F_{r_{j+1}}(Q) & \text{if } \tilde{q} \in Q_{r_j}^{R} \quad (4.14b) \\ F_{r_{j+1}}(\tilde{q}) & \text{if } \tilde{q} \in Q_{r_j}^{P} \quad (4.14c) \end{cases}$$

Using the equations (4.13), (4.14a), (4.14b), and (4.14c), the expected recourse cost in the first direction (i.e., $\delta = 1$) is as follows,

$$\mathcal{Q}^{r,1} = F_{r_1}(Q). \tag{4.15}$$

Where $F_{r_1}(Q)$ is the expected recourse cost of route r, in which the vehicle starts from depot with a full capacity Q, and is computed recursively. Finally, to evaluate the expected recourse cost of the route for the second orientation (i.e., $\mathcal{Q}^{r,2}$), one simply needs to reverse the order of the vertices of the route and reapply the logic of equation (4.15).

4.3 The Integer *L*-shaped Algorithm

We use the integer *L*-shaped algorithm for solving the vehicle routing problem with stochastic demands under the hybrid recourse policy, which was described in the previous section. The integer *L*-shaped algorithm was first proposed by Laporte and Louveaux [34] to solve stochastic programs with binary first-stage variables. This algorithm is an extension of the *L*-shaped algorithm proposed by Van Slyke and Wets [56] for continues stochastic programs, which itself was based on the application of Benders decomposition to stochastic programming, see Benders [3]. In Section §4.3.1 we briefly present the integer *L*-shaped algorithm. Similar to Jabali et al. [28], we use a series of lower bounding functionals (LBFs) based on general partial routes. In section §4.3.2 we present the concept of general partial routes and we present the structure of the LBFs. We note that section §4.3.2 is largely based on Jabali et al. [28], and is presented in this paper for the sake of completeness. In section §4.3.3 we develop bounds specific to our hybrid recourse policy, which are used in the LBFs.

4.3.1 A Brief Description of Integer L-shaped Algorithm

The integer L-shaped algorithm for the VRPSD uses a branch-and-cut scheme, according to which constraints (4.4) and (4.7) are relaxed, the recourse function $Q(x)$ is replaced by a variable Θ, and a general lower bounding constraint (4.16) is applied. Let L denote a general lower-bound value for $Q(x)$, and x is a feasible solution. Then, the initial current problem at iteration $\nu = 0$ is as follows,

$$CP^0 : \min_{x,\Theta} \quad \sum_{i<j} c_{ij} x_{ij} + \Theta \tag{4.1}$$

$$\text{subject to} \quad \sum_{j=2}^{n} x_{1j} = 2m, \tag{4.2}$$

$$\sum_{i<k} x_{ik} + \sum_{k<j} x_{kj} = 2, \qquad k = 2, \ldots, n \tag{4.3}$$

$$0 \leq x_{ij} \leq 1, \qquad 2 \leq i < j < n \tag{4.5}$$

$$0 \leq x_{1j} \leq 2, \qquad j = 2, \ldots, n \tag{4.6}$$

$$L \leq \Theta. \tag{4.16}$$

The algorithm proceeds by adding three types of constraints until optimality is guaranteed: (i) violated constraints (4.4) are gradually added when detected; (ii) valid inequalities

$$L + (\Theta_p - L)W(x) \leq \Theta, \qquad \forall p = \{\alpha, \beta\},\ x \text{ is a partial solution} \tag{4.17}$$

which are elaborated in Section §4.3.3, are added when encountered; and (iii) optimality cuts

$$\sum_{\substack{1 \leq i \leq j \\ x^{\nu}_{ij}=1}} x_{ij} \leq \sum_{1 \leq i \leq j} x^{\nu}_{ij} - 1, \tag{4.18}$$

are added when a feasible integer solution is found to eliminate it from further consideration. We note that the integrality constraints are guaranteed via the branching process. We provide a detailed description of the algorithm in the Appendix (**??**).

The integer L-shaped algorithm was first used by Gendreau et al. [20] to solve the VRP with stochastic demands and customers. Generating all optimality cuts may result in an enumerative process, because each optimality cut solely excludes an integer solution. To counter this effect, researchers have proposed LBF cuts that operate on a large portion of the solution space. Hjorring and Holt [27] proposed LBFs based on partial routes for the single-vehicle routing problem with stochastic demand. LBFs for the multi-VRPSD were proposed by Laporte et al. [35]. Jabali et al. [28] generalized the structure of partial routes to generate several families of LBFs. It is worth noting that since Laporte et al. [35] and Jabali et al. [28] used LBFs for the VRPSD with classical recourse, the bound Θ_p was computed in all cases as defined in Hjorring and Holt [27]. In this paper, we use the LBFs of Jabali et al. [28] for the VRPSD and develop a specific bound Θ_p that is applicable for the proposed hybrid policy.

4.3.2 General Partial Routes

LBFs (4.17) are generated based on partial routes stemming from fractional solutions. In what follows, we define the LBFs using the notation proposed by Jabali et al. [28]. An illustration of a general partial route can be found in Figure (4.1), where the depot is duplicated for presentation convenience. We define $\bar{\mathcal{G}}^v$ as the induced graph by the nonzero variables in the solution of the current problem. We detect partial routes using the exact separation procedure proposed by Jabali et al. [28]. A general partial route is an alternating sequence of the following two components:

1. *Chains* whose vertex set is called chain vertex sets (CVSs). The vertices of a chain are connected to each other by edges (v_i, v_j), for which $x_{ij} = 1$ in $\bar{\mathcal{G}}^v$.
2. *Unstructured components* whose vertex set are called unstructured vertex sets (UVSs).

Each UVS is preceded by a chain and proceeded by another. Each chain is connected to at least one UVS via an *articulation vertex*. In a partial route h, we define ρ as the

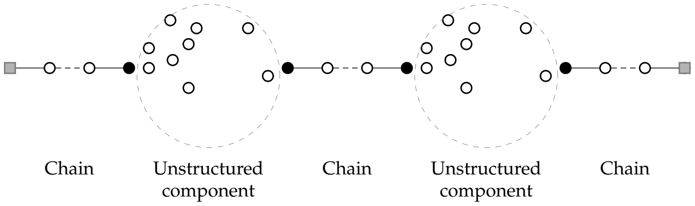


Figure 4.1 – A general partial route h composed of sequenced and unsequenced sets.

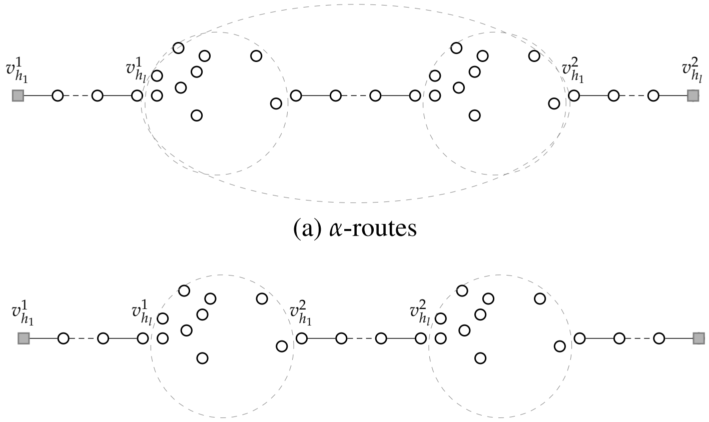
number of chains and $\rho - 1$ as the number of UVSs. Let $S_h^t = \{v_{h_1}^t, \dots, v_{h_l}^t\}$ be the t^{th} chain in partial route h. Therefore, $\sum_{(v_i,v_j)\in S_h^t} x_{ij} = |S_h^t| - 1,\ \forall t = 1, \dots, \rho$. Let U_h^t be the t^{th} UVS in partial route h, then $\sum_{v_i,v_j \in U_h^t} x_{ij} = |U_h^t| - 1,\ \forall t = 1, \dots, \rho - 1$. Ensuring the connectivity of a UVS to the preceding and subsequent chain implies that $\sum_{v_j \in U_h^t} x_{h_l^t j} = 1,\ \forall t \le \rho - 1$ and $\sum_{v_j \in U_h^{t-1}} x_{h_1^t j} = 1,\ \forall t \ge 2$, respectively.

We use two types of partial routes, these are shown Figure (4.2). These types are emerging from the original partial route shown in Figure (4.1), they are denoted by α and β and they are depicted in Figures (4.2a) and (4.2b), respectively. An α-route corresponds to the initial partial route proposed by Hjorring and Holt [27]. The β-route was proposed by Jabali et al. [28]. This type of partial route maintains the exact alternation of CVSs and UVSs.

The functional $W_h(x)$ in LBFs (4.17) was introduced by Jabali et al. [28] for generalized partial routes shown in Figure (4.2), and is defined as follows,

$$\begin{aligned} W_h(x) = {} & \sum_{t=1}^{b} \sum_{\substack{(v_i,v_j)\in S_h^t \\ v_i \neq v_1}} 3x_{ij} + \sum_{(v_1,v_j)\in S_h^1} x_{1j} + \sum_{(v_1,v_j)\in S_h^b} x_{1j} + \sum_{t=1}^{b-1} \sum_{v_i,v_j \in U_h^t} 3x_{ij} \\ & + \sum_{t=1}^{b-1} \sum_{\substack{v_j \in U_h^t \\ v_{h_l}^t \neq v_1}} 3x_{h_l^t j} + \sum_{t=2}^{b} \sum_{\substack{v_j \in U_h^{t-1} \\ v_{h_1}^t \neq v_1}} 3x_{h_1^t j} + \sum_{\substack{v_j \in U_h^1 \\ v_{h_l}^1 = v_1}} x_{h_l^1 j} + \sum_{\substack{v_j \in U_h^{b-1} \\ v_{h_1}^b = v_1 \\ v_{h_1}^{b-1} \neq v_1}} x_{h_1^b j} \\ & - (3|R_h| - 5) \end{aligned} \tag{4.19}$$

The proof of validity of equation (4.17) can be found in Jabali et al. [28]. In the coming



(a) α-routes

(b) β-routes

Figure 4.2 – Generalized partial routes redefined by different views from Figure (4.1).

section we develop the bound Θ_p for the VRPSD with the hybrid policy.

4.3.3 Bounding the Recourse Cost

We now describe the computation of Θ^h_p, which is the lower bound associated with partial route h of type $p \in \{\alpha, \beta\}$. In what follows, we derive the bound for Θ^h_α. This derivation can then be generalized to the computation of Θ^h_β, since this follows a topology containing successive α-route structures.

Let h be a partial route that follows the α topology. Then, one can define h in the following way

$$h = (v_1 = v^1_{h_1}, \dots, v^1_{h_{|S^1_h|}}, U^1_h, v^2_{h_1}, \dots, v^2_{h_{|S^2_h|}} = v_1),$$

where $U^1_h = \{v_{u_1}, v_{u_2}, \dots, v_{u_l}\}$ and $v^1_{h_{|S^1_h|}}$ and $v^2_{h_1}$ are the articulation vertices that connect chains S^1_h and S^2_h to U^1_h. For the sake of simplicity, we redefine the partial route h as

$$h = (v_1 = v_{r_1}, \dots, v_{r_{j-l}}, \{v_{u_1}, v_{u_2}, \dots, v_{u_l}\}, v_{r_{j+1}}, \dots, v_{r_{t+1}} = v_1),$$

where the articulation vertices are relabeled as $v_{r_{j-l}}$ and $v_{r_{j+1}}$. Based on partial route h, we define an artificial route $\tilde{h}$ as follows,

$$\tilde{h} = (v_1 = v_{r_1}, \ldots, v_{r_{j-l}}, \square_{r_{j-l+1}}, \square_{r_{j-l+2}}, \ldots, \square_{r_j}, v_{r_{j+1}}, \ldots, v_{r_{t+1}} = v_1), \qquad (4.20)$$

where $\square_{r_j}$ is the j^{th} position in the artificial route $\tilde{h}$. Positions $\square_{r_{j-l+1}}, \ldots, \square_{r_j}$ could contain any possible permutation of customers in U_h^1. We develop a bounding procedure for the artificial route $\tilde{h}$ which bounds for all possible assignment of customers in U_h^1.

We recall that the expected recourse cost upon arrival at the k^{th} customer in r with q units of residual capacity is computed as follows,

$$F_{r_k}(q) = \mathbb{E}_{\xi_{r_k}}\left[F_{r_k}^{post}(q - \xi_{r_j})\right] = \mathbb{E}_{\xi_{r_k}}\left[F_{r_k}^{post}(\tilde{q})\right],\ \forall \tilde{q} = q - \xi_{r_k}. \qquad (4.13)$$

For the sake of simplicity, we use the notation $F_{r_k}(.)$ as defined in (4.23) whenever it can be exactly applied (namely in the chain the positions of $\tilde{h}$). Therefore, $F_{r_{t+1}}(q)$,...,$F_{r_{j+1}}(q)$ for all q can be exactly computed by recourse function (4.23). Considering positions $k = j - l + 1$ and $k = j$, we denote by $\widetilde{F}_{r_k}(q)$ as the lower bound on the expected recourse cost at the k^{th} position in artificial route h with q units of residual capacity. In Lemma 4.3.1 we bound the onward recourse cost from the j^{th} customer, which can potentially be any of the unsequenced customers in U_h^1. In Lemma 4.3.2 we then bound the onward recourse cost from the $j - l + 1^{\text{th}}$ customer. We recall that

$$F_{r_j}(q) = \mathbb{E}_{\xi_{r_j}}\left[F_{r_j}^{post}(q - \xi_{r_j})\right] = \mathbb{E}_{\xi_{r_j}}\left[F_{r_j}^{post}(\tilde{q})\right],\ \forall \tilde{q} = q - \xi_{r_j}. \qquad (4.13)$$

Lemma 4.3.1. *A lower bound on the expected recourse cost at the j^{th} customer for each q follows:*

$$\widetilde{F}_{r_j}(q) = \min_{v_{u_e} \in U_h^1} F_{r_j}(q)|_{r_j := u_e}, \qquad (4.21)$$

where $F_{r_j}(q)|_{r_j := u_e}$ can be computed by accounting for v_{u_e} as the j^{th} customer in the recourse function (4.13).

Proof. Since the j^{th} customer is unsequenced, it can potentially be any $v_{u_e} \in U_h^1$. We bound the onward expected recourse cost at the j^{th} customer by minimizing the recourse cost over the unsequenced set for each q. Then, $\widetilde{F}_{r_j}(.) \leq F_{r_j}(.)|_{r_j := u_e}$ by definition. □

Lemma 4.3.2. *The lower bound on $\widetilde{F}_{r_{j-l+1}}(.)$ for each q can be directly obtained as follows*

$$\widetilde{F}_{r_{j-l+1}}(q) = \min_{U \subset U_h^1 : |U| = |U_h^1| - 1} \prod_{v_{u_e} \in U} p_{u_e}^{l^{min}} . \widetilde{F}_{r_j}(q), \tag{4.22}$$

where, $p_i^{l^{min}} = \min\{p_i^1, \ldots, p_i^{\xi_i^{s^i}}\}$.

Proof. By definition, $\min\limits_{U \subset U_h^1 : |U| = |U_h^1| - 1} \prod\limits_{v_{u_e} \in U} p_{u_e}^{l^{min}}$ is a lower bound on the probability of the stochastic events that occur at the $j - 1^{\text{th}}$ customer. Since $\widetilde{F}_{r_j}(q)$ is lower bound on the expected recourse cost at the j^{th} customer (as shown in Lemma 4.3.1), Equation (4.22) is a lower bound on the excepted recourse cost of the $j - l + 1^{\text{th}}$. □

Using the bounds specified in Lemma 4.3.1 and Lemma 4.3.2, the recourse function (4.23) can be slightly modified to compute $F_{r_{j-l}}^{post}(\tilde{q})$ can be expressed as follows.

$$F_{r_{j-l}}^{post}(\tilde{q}) = \begin{cases} b + 2c_{1r_{j-l}} + \widetilde{F}_{r_{j-l+1}}(Q + \tilde{q}) & \text{if } \tilde{q} < 0 \\ \tilde{c}_{r_{j-l}} + \widetilde{F}_{r_{j-l+1}}(Q) & \text{if } \tilde{q} \in Q_{r_j}^R \\ \widetilde{F}_{r_{j-l+1}}(\tilde{q}) & \text{if } \tilde{q} \in Q_{r_j}^P, \end{cases}$$

where, $\tilde{c}_{r_{j-l}} = \underset{v_{u_e} \in U_h^1}{minimum}\{c_{1,r_{j-l}} + c_{1,u_e} - c_{r_{j-l},u_e}\}$. The above computation enables the computation of $F_{r_{j-l}}(.)$ by equation (4.13). The excepted recourse cost of the remaining positions can therefore be successively computed as $F_{r_{j-l-2}}(.), \ldots, F_{r_1}(.)$ using recourse function (4.23). Ultimately $F_{r_1}(Q)$ bounds the expected recourse cost of artificial route h. This bound is computed for both orientations of the partial route and the minimum value is assumed to be the lower bound Θ_α^h. We recall that the mechanism for computing Θ_α^h is reapplied to compute Θ_β^h, where the latter is treated as a succession of

α-route structures. In the LBF cuts (4.17) the bound Θ_p is decomposed by partial routes (or routes) as $\Theta_p = \sum_{r=1}^{m} \Theta_p^r$, where $p = \{\alpha, \beta\}$.

4.4 Numerical Experiments

The first aim of this section is to demonstrate the effectiveness of the solution algorithm on a large set of experiments. The second aim is to verify the added value of using the proposed hybrid policy when compared to other polices. In what follows, we detail the instance generation, the performance of the algorithm is verified in Section 4.4.1, while a comparison with the other policies is performed in Section 4.4.2.

We use the instances of Salavati-Khoshghalb et al. [45], for completeness we briefly describe the instance generation procedure. For each instance, a set of $V = \{v_1, \ldots, v_n\}$ (where v_1 is the depot) is generated in a $[0, 100]^2$ square following a continuous uniform distribution. The travel costs are then set to the nearest integer associated to the Euclidean distance between two vertices. Each customer is randomly (with equal probability) selected to have low, medium, or high demand. These three classifications correspond to ranges $[1, 5]$, $[6, 10]$, and $[11, 15]$, respectively. For the selected range, the demand realizations are randomly generated for each customer with probabilities $\{0.1, 0.2, 0.4, 0.2, 0.1\}$, corresponding to the five values in the range. We consider 11 pairs of (n, m) as indicated in Table 4.I, we recall that m denotes the number of vehicles. Four fill rate coefficients are considered for each of the 11 combinations, where the fill rate is computed as $\bar{f} = \frac{\sum_{i=2}^{n} \mathbb{E}(\xi_i)}{mQ}$. The capacity of each vehicle Q is inferred from $\bar{f}$. The cost b is set to $\sum\limits_{i=2,\ldots,n} c_{i1}/(n-1)$, which is the average distance to the depot when considering all customers. Furthermore, L is set to zero. For each combination in Table 4.I, ten instances were generated, thus yielding a total of 440 instances.

Table 4.I – 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

We chose five pairs of values for the maximum proceeding threshold $\underline{\theta}$ and the minimum restocking threshold $\bar{\theta}$. Each pair $\{\underline{\theta}, \bar{\theta}\}$ is chosen as $\{0.5 - \lambda, 0.5 + \lambda\}$, where λ takes one of the following values $\{0.05, 0.15, 0.25, 0.35, 0.45\}$. Thus, the following five pairs are used $\{\underline{\theta}, \bar{\theta}\}$: $\{0.45, 0.55\}$, $\{0.35, 0.65\}$, $\{0.25, 0.75\}$, $\{0.15, 0.85\}$, and $\{0.05, 0.95\}$. Each instance is solved considering each of the five pairs, thus yielding a total of 2200 experiments.

The algorithm is coded in C++ using ILOG CPLEX 12.6. All experiments were performed, using a single thread, on a cluster of 27 computers, each of which having 12 cores, two Intel(R) Xeon(R) X5675 3.07 GHz processors and 96 GB of RAM. The branching was managed by the OOBB package of [24]. The separation problem of constraints (4.4) is solved using the CVRPSEP package of [38]. The maximum CPU time limit is set to 10 hours and the optimality gap was set to 0.01%.

4.4.1 Results for the hybrid recourse policy

The performance of the exact algorithm for the hybrid policy is presented in Tables 4.II-4.VI with the five pairs of values $\{\underline{\theta}, \bar{\theta}\}$, each corresponding to a table. Column “solved" expresses the number of optimally solved instances (out of ten), column “Run (sec)" reports the average run time of those solved instances and column “Gap" reports the average gap on all instances.

The total number of optimally solved instances for each of the five pairs of $\{\underline{\theta}, \bar{\theta}\}$ are 281, 283, 282, 279 and 279, out of 440. Overall our algorithm solved between 60.2% and

64.3% of the instances to optimality. These results are rather competitive for the SVRP literature, see Gendreau et al. [22] for further details. The weighted average time (in seconds) to solve an instance to optimality for the four $\bar{f}$ values are: 1332.29, 1274.63, 1549.79, and 1205.95. The total average gaps over the four $\bar{f}$ values are computed for each pair of $\{\underline{\theta},\bar{\theta}\}$ are 0.50%, 0.50%, 0.53%, 0.55% and 0.55%.

Considering a fill rate of 0.90 and the five pairs of $\{\underline{\theta},\bar{\theta}\}$, Tables 4.II-4.VI show that our algorithm was able to solve between 84 and 87 instances (from the total of 110). Instances with up to 60 nodes are solved to optimality. Considering a fill rate of 0.96 and the five pairs of $\{\underline{\theta},\bar{\theta}\}$, our algorithm was able to solve between 37 and 42 instances. However, the overall obtained gaps are relatively small, with the largest average gap being 1.38%, as reported in Table 4.VI .

The results in Tables 4.II-4.VI also indicate that the problems become harder to solve with the increase in fill rate, number of vehicle and number of nodes. These results are consistent with the findings of both Laporte et al. [35] and Jabali et al. [28]. Finally, the number of solved instances to optimality varies only slightly over the pairs of $\{\underline{\theta},\bar{\theta}\}$. Thus, indicating that the proposed algorithm remains robust even when the values defining the policy vary.

Table 4.II – Hybrid policy with $\{\underline{\theta},\bar{\theta}\}$=$\{0.45, 0.55\}$

n	m	$\bar{f}$	solved	Run(sec)	Gap	$\bar{f}$	solved	Run(sec)	Gap	$\bar{f}$	solved	Run(sec)	Gap	$\bar{f}$	solved	Run(sec)	Gap
20	2	0.90	10	0.80	0.00%	0.92	10	1.40	0.00%	0.94	10	0.50	0.00%	0.96	10	52.10	0.00%
30	2	0.90	10	0.10	0.00%	0.92	10	19.90	0.00%	0.94	10	60.60	0.00%	0.96	8	3065.00	0.12%
40	2	0.90	10	1.20	0.00%	0.92	10	2.60	0.00%	0.94	10	5.60	0.00%	0.96	6	90.17	0.13%
40	3	0.90	10	2023.80	0.01%	0.92	8	216.38	0.20%	0.94	8	2416.38	0.06%	0.96	5	14046.40	1.10%
40	4	0.90	5	1434.20	0.70%	0.92	2	15054.50	2.08%	0.94	1	2600.00	1.63%	0.96			4.28%
50	2	0.90	10	3.40	0.00%	0.92	10	78.20	0.00%	0.94	10	11.00	0.01%	0.96	4	222.25	0.11%
50	3	0.90	9	2998.56	0.22%	0.92	7	5886.29	0.56%	0.94	10	1501.30	0.01%	0.96	1	5.00	2.10%
50	4	0.90	2	1.00	0.63%	0.92	2	16666.50	1.02%	0.94	2	3369.50	1.80%	0.96			3.00%
60	2	0.90	10	488.40	0.00%	0.92	10	8.80	0.00%	0.94	9	701.78	0.02%	0.96	7	427.14	0.02%
60	3	0.90	7	707.71	0.35%	0.92	7	700.29	0.57%	0.94	5	3051.60	0.63%	0.96	1	19738.00	0.57%
60	4	0.90	2	345.00	1.30%	0.92	1	6554.00	1.36%	0.94	2	2491.00	1.30%	0.96			3.99%
Average				764.48	0.21%			1544.70	0.39%			922.29	0.36%			2843.71	1.03%
Total			85				77				77				42		

Table 4.III – Hybrid policy with $\{\underline{\theta},\bar{\theta}\}$={0.35, 0.65}

n	m	$\bar{f}$	solved	Run(sec)	Gap	$\bar{f}$	solved	Run(sec)	Gap	$\bar{f}$	solved	Run(sec)	Gap	$\bar{f}$	solved	Run(sec)	Gap
20	2	0.90	10	0.70	0.00%	0.92	10	1.20	0.00%	0.94	10	0.50	0.00%	0.96	10	47.50	0.00%
30	2	0.90	10	0.10	0.00%	0.92	10	16.10	0.00%	0.94	10	48.40	0.00%	0.96	8	2994.00	0.11%
40	2	0.90	10	1.20	0.00%	0.92	10	2.20	0.00%	0.94	10	4.70	0.00%	0.96	6	81.00	0.12%
40	3	0.90	10	1918.50	0.01%	0.92	8	172.88	0.19%	0.94	8	2015.62	0.06%	0.96	5	10656.40	1.07%
40	4	0.90	5	1594.80	0.68%	0.92	2	10777.50	1.97%	0.94	1	1714.00	1.58%	0.96			4.20%
50	2	0.90	10	3.10	0.00%	0.92	10	65.70	0.00%	0.94	10	10.20	0.01%	0.96	4	195.75	0.09%
50	3	0.90	9	1785.67	0.21%	0.92	7	4983.00	0.53%	0.94	10	1253.00	0.01%	0.96	1	5.00	2.02%
50	4	0.90	3	8875.33	0.59%	0.92	2	14213.00	1.01%	0.94	2	3069.50	1.76%	0.96			2.99%
60	2	0.90	10	355.40	0.00%	0.92	10	8.20	0.00%	0.94	9	701.67	0.01%	0.96	7	363.29	0.01%
60	3	0.90	8	4653.50	0.27%	0.92	7	589.86	0.48%	0.94	5	2505.40	1.19%	0.96	1	11834.00	0.54%
60	4	0.90	2	321.00	1.20%	0.92	1	4914.00	1.30%	0.94	2	1909.50	1.29%	0.96			4.03%
Average				1279.67	0.20%			1249.64	0.37%			776.71	0.39%			2222.86	1.01%
Total			87				77				77				42		

Table 4.IV – Hybrid policy with $\{\underline{\theta},\bar{\theta}\}$={0.25, 0.75}

n	m	$\bar{f}$	solved	Run(sec)	Gap	$\bar{f}$	solved	Run(sec)	Gap	$\bar{f}$	solved	Run(sec)	Gap	$\bar{f}$	solved	Run(sec)	Gap
20	2	0.90	10	0.90	0.00%	0.92	10	2.40	0.00%	0.94	10	0.60	0.00%	0.96	10	83.30	0.01%
30	2	0.90	10	0.10	0.00%	0.92	10	16.40	0.00%	0.94	10	106.10	0.00%	0.96	8	3101.12	0.14%
40	2	0.90	10	1.00	0.00%	0.92	10	2.50	0.00%	0.94	10	5.20	0.00%	0.96	6	69.33	0.11%
40	3	0.90	10	2429.60	0.01%	0.92	8	219.88	0.21%	0.94	8	2437.38	0.06%	0.96	5	11274.80	1.14%
40	4	0.90	5	1965.80	0.86%	0.92	1	28952.00	2.29%	0.94	1	10841.00	1.77%	0.96			4.63%
50	2	0.90	10	2.90	0.00%	0.92	10	124.20	0.00%	0.94	10	11.40	0.01%	0.96	5	6864.60	0.06%
50	3	0.90	9	2404.11	0.18%	0.92	7	4533.00	0.47%	0.94	10	1114.60	0.01%	0.96	1	3.00	2.04%
50	4	0.90	3	11876.67	0.68%	0.92	1	405.00	1.10%	0.94	2	6434.50	1.90%	0.96			3.28%
60	2	0.90	10	363.30	0.00%	0.92	10	7.40	0.00%	0.94	9	703.11	0.01%	0.96	7	305.71	0.02%
60	3	0.90	8	4368.12	0.27%	0.92	7	625.29	0.50%	0.94	5	7206.20	1.20%	0.96	1	13311.00	0.51%
60	4	0.90	2	293.00	1.46%	0.92	1	4882.00	1.41%	0.94	2	1322.00	1.35%	0.96			4.27%
Average				1501.21	0.23%			981.80	0.40%			1306.38	0.42%			3074.63	1.08%
Total			87				75				77				43		

Table 4.V – Hybrid policy with $\{\underline{\theta},\bar{\theta}\}$={0.15, 0.85}

n	m	$\bar{f}$	solved	Run(sec)	Gap	$\bar{f}$	solved	Run(sec)	Gap	$\bar{f}$	solved	Run(sec)	Gap	$\bar{f}$	solved	Run(sec)	Gap
20	2	0.90	10	1.00	0.00%	0.92	10	2.40	0.00%	0.94	10	0.60	0.00%	0.96	10	77.40	0.01%
30	2	0.90	10	0.10	0.00%	0.92	10	15.80	0.00%	0.94	10	96.40	0.00%	0.96	8	2707.62	0.14%
40	2	0.90	10	1.10	0.00%	0.92	10	2.60	0.00%	0.94	10	5.20	0.00%	0.96	6	67.83	0.11%
40	3	0.90	10	2176.10	0.01%	0.92	8	208.75	0.22%	0.94	8	2210.50	0.06%	0.96	5	11499.40	1.14%
40	4	0.90	5	1898.60	0.87%	0.92	1	25147.00	2.31%	0.94	1	11220.00	1.79%	0.96			4.66%
50	2	0.90	10	3.20	0.00%	0.92	10	103.60	0.00%	0.94	10	11.30	0.01%	0.96	4	227.00	0.09%
50	3	0.90	9	2711.22	0.22%	0.92	7	4878.43	0.47%	0.94	10	1194.90	0.01%	0.96	1	4.00	2.09%
50	4	0.90	3	10126.33	0.61%	0.92	1	388.00	1.12%	0.94	2	7366.50	1.92%	0.96			3.29%
60	2	0.90	10	364.50	0.00%	0.92	10	7.80	0.00%	0.94	9	592.78	0.02%	0.96	7	270.43	0.02%
60	3	0.90	7	448.14	0.29%	0.92	7	589.00	0.58%	0.94	4	2182.00	1.21%	0.96	1	14918.00	0.53%
60	4	0.90	2	297.50	1.53%	0.92	1	5012.00	1.71%	0.94	2	1168.50	1.38%	0.96			4.46%
Average				1086.80	0.24%			957.48	0.43%			962.12	0.43%			2334.81	1.10%
Total			86				75				76				42		

Table 4.VI – Hybrid policy with $\{\underline{\theta},\bar{\theta}\}$=$\{0.05, 0.95\}$

n	m	$\bar{f}$	solved	Run(sec)	Gap	$\bar{f}$	solved	Run(sec)	Gap	$\bar{f}$	solved	Run(sec)	Gap	$\bar{f}$	solved	Run(sec)	Gap
20	2	0.90	10	1.70	0.00%	0.92	10	10.00	0.00%	0.94	10	2.70	0.00%	0.96	10	895.00	0.01%
30	2	0.90	10	0.30	0.00%	0.92	10	16.50	0.00%	0.94	10	3397.10	0.00%	0.96	7	2708.00	0.26%
40	2	0.90	10	1.90	0.00%	0.92	10	3.20	0.00%	0.94	10	7.20	0.00%	0.96	6	126.67	0.18%
40	3	0.90	9	2530.44	0.05%	0.92	8	920.62	0.32%	0.94	6	7261.67	0.17%	0.96	2	14351.50	1.73%
40	4	0.90	5	11628.40	1.22%	0.92	1	16149.00	3.36%	0.94			2.84%	0.96			5.85%
50	2	0.90	10	2.90	0.00%	0.92	10	559.50	0.00%	0.94	10	16.00	0.00%	0.96	4	2173.25	0.17%
50	3	0.90	9	2998.11	0.22%	0.92	6	4342.33	0.53%	0.94	9	2128.78	0.04%	0.96	1	6.00	2.42%
50	4	0.90	2	2.00	1.03%	0.92	1	4991.00	1.54%	0.94			2.50%	0.96			4.18%
60	2	0.90	10	422.80	0.00%	0.92	10	8.40	0.00%	0.94	9	571.89	0.02%	0.96	7	380.71	0.06%
60	3	0.90	7	1149.00	0.32%	0.92	7	1277.00	0.56%	0.94	4	5015.25	1.40%	0.96			0.72%
60	4	0.90	2	778.50	1.54%	0.92	1	12358.00	2.17%	0.94	2	16931.50	1.65%	0.96			5.16%
Average				1449.99	0.29%			1105.84	0.57%			2229.00	0.58%			1857.65	1.38%
Total			84				74				70				37		

4.4.2 Recourse cost analysis

The objective of this section is to analyze the hybrid risk-and-distance-based policy with respect to other policies. To do so, we focus the analyses on the instances solved to optimality. We initially compare our policy with the classical one by evaluating the routes associated with the solutions obtained using the hybrid policy under the classical policy.

We first compare the expected number of recourse actions taken in the classical recourse policy when compared with its counterpart (i.e., the hybrid policy). We recall that the recourse actions in the classical recourse policy are back-and-forth trips and restocking trips. Based on the results obtained when applying the classical policy, we computed the expected number of back-and-forth trips EBF_c and the expected number of restocking trips ER_c. Thus, the total expected number of recourse actions when applying the classical recourse policy to the considered routes is expressed as $EBF_c + ER_c$. As for the hybrid policy, the recourse actions are back-and-forth trips and preventive restocking trips. As previously mentioned, in this policy, an exact stock out triggering a restocking trip is considered as a preventive restocking trip. Therefore, for the hybrid policy, we computed the expected number of back-and-forth trips EBF_h and the expected number of preventive restocking trips EPR_h. Thus, the total expected number of recourse actions in the hybrid recourse policy is expressed as $EBF_h + EPR_h$.

In Table 4.VII, we report the average ratio between expected number of recourse actions between the hybrid policy and the classical policy. We observe that the expected number of recourse actions is higher for the hybrid policy, when compared to the classical policy. This tendency increases with $\{\underline{\theta}, \bar{\theta}\}$ and is relatively consistent through the varying values of $\bar{f}$. These results could be interpreted by the hybrid policy being more risk averse than the classical one, and thus prescribes more recourse actions. However, as we will see next, the expected number of BF trips are reduced when using the hybrid policy. Moreover, the final analysis of this section shows that the hybrid policy yields less costly solutions, when compared to the classical policy.

Table 4.VII – The ratio $\dfrac{EBF_c + EPR_c}{EBF_h + ER_h}$

	$\bar{f}$			
($\underline{\theta}$-$\bar{\theta}$)	0.90	0.92	0.94	0.96
$0.45 - 0.55$	88.45%	88.31%	88.29%	88.86%
$0.35 - 0.65$	88.74%	88.31%	88.29%	88.86%
$0.25 - 0.75$	65.23%	65.07%	65.87%	68.87%
$0.15 - 0.85$	65.24%	65.07%	65.87%	68.75%
$0.05 - 0.95$	43.25%	42.90%	45.30%	50.18%

We now focus on the expected number of back-and-forth trips performed by the hybrid policy and the classical policy, i.e., EBF_h and EBF_c. This analysis is important since back-and-forth trips imply a disruption at the customer location, thus EBF_h and EBF_c reflect a measure of customer service. In Table 4.VIII, we report the ratio between EBF_c and EBF_h. We clearly observe that this ratio is largely impacted by the values defining the hybrid policy $\{\underline{\theta}, \bar{\theta}\}$. We note that the last line of the table is empty since no BF trips are preformed under the hybrid policy with $\{\underline{\theta}, \bar{\theta}\} = \{0.05, 0.95\}$. This large interval implies that resulting policy is rather conservative.

Table 4.VIII – The ratio $\frac{EBF_c}{EBF_h}$

$\underline{\theta}$-$\bar{\theta}$	$\bar{f} = 0.90$	$\bar{f} = 0.92$	$\bar{f} = 0.94$	$\bar{f} = 0.96$
$0.45 - 0.55$	3.49	3.72	3.85	4.54
$0.35 - 0.65$	3.50	3.72	3.85	4.54
$0.25 - 0.75$	10.46	11.43	11.78	14.47
$0.15 - 0.85$	10.47	11.43	11.79	14.37
$0.05 - 0.95$	—	—	—	—

As observed from the previous analysis, preventive returns in the hybrid recourse policies hedge the occurrence of route failures. However, this could result in extra recourse cost being incurred. In order to evaluate the quality of the rule-based policies presented in this paper in terms of the incurred recourse cost, the optimal solutions obtained with the hybrid policy are priced under both the classical and optimal restocking policies. Let x denote the optimal solution obtained with the hybrid policy, the first stage cost is cx, let $Q^h(x)$, $Q^c(x)$, and $Q^o(x)$ express the expected recourse cost of x with the hybrid, classical and optimal restocking policies, respectively. Where $Q^o(x)$ was computed using a similar approach as the one presented in Bertsimas et al. [5]. Two cost measures are used to assess the results obtained, "Savings" $= \frac{Q^c(x)-Q^h(x)}{cx+Q^c(x)}$ and "Deviations" $= \frac{Q^h(x)-Q^o(x)}{cx+Q^o(x)}$.

Table 4.IX summarizes the average results on the savings and the deviations. The values in this table are generally low. This is to be expected since, in the VRPSD, the first stage cost tends to dominate the recourse cost. Such observations are consistent with the findings reported in the VRPSD literature (e.g., Bianchi [6] and Rei et al. [44]). We note that the hybrid policy yields a positive average savings on all entries of the table. The maximum average saving is 1.19% for the combination of $\{\underline{\theta}, \bar{\theta}\} = \{0.25, 0.75\}$ with $\bar{f} = 0.96$. The savings tend to increase with the fill rate, this can be explained by the reduction of the expected number of failures observed in Table 4.VIII.

Comparing the costs of the hybrid policy with those of the optimal restocking policy we observe that the deviations are rather small. Thus implying that for the considered routes, the use of the hybrid policy scales well compared to the optimal restocking one. Overall, for the considered routes, one can conclude that the opportunity loss of not implementing the optimal policy is very low. Furthermore, the hybrid policy seems to provide a very good approximation of the optimal one.

Table 4.IX – Savings and Deviations.

	$\bar{f} = 0.90$		$\bar{f} = 0.92$		$\bar{f} = 0.94$		$\bar{f} = 0.96$	
$\underline{\theta}$-$\bar{\theta}$	Savings	Deviations	Savings	Deviations	Savings	Deviations	Savings	Deviations
0.45 – 0.55	0.13%	0.01%	0.19%	0.01%	0.39%	0.02%	0.47%	0.02%
0.35 – 0.65	0.13%	0.01%	0.19%	0.01%	0.39%	0.02%	0.47%	0.02%
0.25 – 0.75	0.11%	0.02%	0.18%	0.02%	0.37%	0.04%	1.19%	0.07%
0.15 – 0.85	0.11%	0.02%	0.18%	0.02%	0.37%	0.04%	1.19%	0.07%
0.05 – 0.95	0.03%	0.08%	0.08%	0.09%	0.22%	0.20%	0.95%	0.35%

4.5 Conclusions

In this paper, we have defined a general taxonomy to classify rule-based recourse policies for the VRPSD. According to this taxonomy, rule-based polices are cast into three general classes. We introduced the first hybrid policy, which simultaneously combines two of these classes, namely risk and distance. We modelled the VRPSD with the hybrid risk-and-distance-based policy and derived the computations of the resulting recourse cost. Furthermore, we proposed an exact solution algorithm, for which we developed bounds that are used in the LBFs.

The exact algorithm was able to solve a large number of instances to optimality, especially for low fill rates. For example, considering a fill rate of 0.90 with $\{\underline{\theta}, \bar{\theta}\} = \{0.35, 0.65\}$, up to 79% of the instances were solved to optimality. The algorithm also scales well in terms of the sizes of the instances, it solved to optimality instances with up to 60 nodes. Furthermore, the average observed optimality gaps are rather low.

Through our experimental study, we observed that the expected number of failures are noticeably lower when applying the hybrid policy compared to the classical policy. These results indicate the superiority of the hybrid policy in terms of customer service. We further observed that the optimal solutions of the hybrid policy yield cost savings when compared to the classical policy. Finally, we also showed that the cost offset of the optimal restocking policy compared to the hybrid one is rather small.

Appendix

The Integer L-shaped Algorithm

We briefly describe here the Integer *L*-shaped algorithm (0). As a branch-and-cut algorithm, first, we state the initial current problem (CP) with relaxing the capacity / subtour-elimination constraints (4.4), and integrality constraints (4.7).

The integer *L*-shaped algorithm (0) in **Step 0** sets the iteration index, the overall upper bound, and pushes the initial CP as the first pendant node. In **Step 1**, the algorithm checks the pendant list for any pendant node available, if not applicable then `stop`. In **Step 2**, the algorithm solves the pendant CP optimally. The algorithm checks for any violation of capacity constraints (4.4) in **Step 3**, and generates in case associated constraints, and adds the updated subproblem to the pendant list. In addition, the associated LBFs will be added to improve the lower bound of expected recourse cost.

Also, the algorithm checks integrality constraints (4.7) in **Step 4**. If the optimal solution is non-integer, then branching procedure adds new updated CPs to the pendant list. Otherwise, an integer solution is obtained, and the algorithm computes the expected recourse cost of optimal routing solution. Since an integer solution is obtained, the algorithm checks to update the overall upper bound in **Step 5**. Then, the algorithm checks for an excessive expected recourse cost to add optimality cuts in in **Step 6**.

```
▷ state initial $CP$ with the constraints: 4.2, 4.3, 4.5, 4.6, and $L \leq \Theta$.
▷ Step 0: set iteration index and initial upper bound
$\nu \leftarrow 0$
$\bar{z} \leftarrow +\infty$
push the initial $CP$ in the list of pendant nodes, $list_{PN}$.
▷ Step 1: check search tree for a pendant node
if $list_{PN}$ is empty then
    STOP
end if
▷ Step 2: increase iteration index, and solve $CP$ optimally
```

```
$\nu \leftarrow \nu + 1$
let $(x^{\nu}, \Theta^{\nu})$ is the optimal solution of $CP$
▷ **Step 3**: check for any violation of (4.4).
**if** There are any such violated constraint **then**
    generate associated cuts and LBFs and add them to $CP$
    go to **Step 2**
**else if** $cx^{\nu} + \Theta^{\nu} \geq \bar{z}$ **then**
    fathom the current node
    go to **Step 1**
**end if**
▷ **Step 4**: check for any integrity violation.
**if** there are any such violated constraints **then**
    generate the branching subproblems and append to pendant list $list_{PN}$
    go to **Step 2**
**end if**
▷ **Step 5**: check for a new integer incumbent.
compute $\mathcal{Q}(x^{\nu})$
$z^{\nu} \leftarrow cx^{\nu} + \mathcal{Q}(x^{\nu})$
**if** $z^{\nu} < \bar{z}$ **then**
    $\bar{z} \leftarrow z^{\nu}$
**end if**
▷ **Step 6**: check for optimality cuts.
**if** $\Theta^{\nu} \geq \mathcal{Q}(x^{\nu})$ **then**
    fathom the current node
    go to **Step 1**
**else**
    add an optimality cut
        $$\sum_{\substack{1 \leq i \leq j \\ x^{\nu}_{ij} = 1}} x_{ij} \leq \sum_{1 \leq i \leq j} x^{\nu}_{ij} - 1 \qquad (4.24)$$
**end if**
```

CHAPTER 5

ARTICLE 3: AN EXACT ALGORITHM TO SOLVE THE VEHICLE ROUTING PROBLEM WITH STOCHASTIC DEMANDS UNDER AN OPTIMAL RESTOCKING POLICY

This chapter will be submitted to the *European Journal of Operational Research*. Preliminary work was presented at the following conferences:

- CMS 2017, Bergamo, Italy, May 30-31, June 1, 2017
- Optimization Days 2017, Montreal, Canada, May 10-11, 2017
- EURO 2016, Poznan, Poland, July 3-7, 2016

Contribution:

- The general orientations of paper are proposed by the supervisors.
- Conducting the research including the modelling of ideas, implementations and coding parts is carried out by student. The draft versions of the three papers are written by student and then are modified by supervisors.

An Optimal Restocking Recourse Policy for the Vehicle Routing Problem with Stochastic Demands

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

[1]Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport (CIRRELT), C.P. 6128, succ. Centre-ville, Montréal, Québec H3C 3J7, Canada
[2]Département d'informatique et de recherche operationnelle, Université de Montréal, C.P. 6128, succ. Centre-ville, Montréal, Québec H3C 3J7, Canada
[3]Département de mathématiques et de génie industriel, Polytechnique de Montréal, C.P. 6079, succ. Centre-ville, Montréal, Québec H3C 3J7, Canada
[4]Dipartimento di elettronica, informazione e bioingegneria, Politecnico di Milano, Piazza Leonardo da Vinci, 32, Milano 20133, Italy
[5]Département de management et technologie, École des sciences de la gestion, Université du Québec à Montréal, C.P. 8888, succ. Centre-ville, Montréal, Québec H3C 3P8, Canada

Abstract

This paper examines the Vehicle Routing Problem with Stochastic Demands (VRPSD), in which the actual demand of customers can only be realized upon arriving at the customer location. Under demand uncertainty, a planned route may fail at a specific customer when the observed demand exceeds the residual capacity. There are two ways to face such failure events, at which the vehicle can either execute a return trip to the depot at the failure location and refill the capacity and complete the split service, or in anticipation of potential failures perform a preventive return whenever the residual capacity falls below a threshold; overall, these return trips are called recourse actions. In the context of VRPSD, a recourse policy which schedules various recourse actions based on the visits planned beforehand on the route must be designed. An optimal recourse policy prescribes the cost-effective returns based on a set of optimal customer-specific thresholds. We propose an exact solution method to solve the multi-VRPSD under an optimal restocking policy. The integer *L*-shaped algorithm is adapted to solve the VRPSD in a branch-and-cut framework. To enhance the efficiency of presented algorithm, several lower bounding schemes are redeveloped by approximating the expected recourse cost.

Keywords: vehicle routing problem; stochastic demands; optimal policy; restocking; partial routes; L-shaped algorithm; lower bounding functionals.

5.1 Introduction

Following the seminal paper of Dantzig and Ramser [15], the *Vehicle Routing Problem* (VRP) has been the subject of considerable research efforts over the last decades, see Laporte [32]. The aim in VRP is to find a set of routes serving all customers in a govern set at a minimal cost (the least travel cost, minimum number of vehicles, etc.). The routes should start and end at the depot, and are designed to be performed by a fleet of vehicles with homogeneous capacity. In the deterministic version of VRP in which all problem parameters are known precisely, each customer is only visited once by one vehicle.

In real-life problems, however, various parameters of the VRP can be uncertain. Uncertainty is more likely to appear in demands, travel and service times, and customer presence. It is usually dealt with by using probability distributions to describe the uncertain parameters, which are then stochastic. The VRPs in which some parameters are stochastic are called *Stochastic* VRPs (SVRPs). Although SVRPs have received much less attention in comparison to the deterministic VRP, several efforts have been devoted to investigate various versions of the SVRP; for a thorough exposition of the SVRP context, we refer the reader to Gendreau et al. [22]. One way to deal with stochastic parameters in stochastic routing models is to use their deterministic approximated counterparts, in which the stochastic parameters are roughly replaced by their forecasted equivalents. Such models can sometimes lead to arbitrarily bad quality solutions at execution time when stochasticity reveals itself, see Louveaux [36]. Thus, there is a need to model SVRPs using specialized optimization frameworks in which stochastic parameters are explicitly modeled through random variables.

In this paper, we are mainly interested in the *Vehicle Routing Problem with Stochastic Demands* (VRPSD), where customer demands are only known through probability distributions. In this context, it is common to assume that the actual demand of each customer can only be observed upon arriving at its location. Because of that, a planned route may *fail* at a customer when the demand exceeds the residual capacity on the vehicle. This occurrence is called a *route failure*. To prevent failures and complete the service after a route failure has occurred, extra decisions, called *recourse actions*, must be taken and associated travel costs, called recourse costs, need to be incurred. The objective in the VRPSD is to minimize the total driven distance, which consists of routing (i.e., preliminary plans) costs and recourse costs.

It is important to note that the general context of the VRPSD can be tackled in variety of ways. One thus usually refers to *modeling paradigms* to formalize the problem and the way in which it is solved. Dror et al. [17] describe several of these paradigms for the VRPSD. One of them is the so-called *a priori optimization* approach, which was extensively discussed in Bertsimas et al. [4]; another is the *reoptimization* approach; further details can be found in Gendreau et al. [22]. These modeling paradigms either separate or unify the process of making routing and recourse decisions, where information, here, stochastic demands, are revealed at once or in a stepwise manner, respectively. In the a priori optimization approach, one decomposes the overall decision making process into two sets of mutually exclusive decisions as routing and recourse decisions, thus modeling the VRPSD as a two-stage stochastic integer program with recourse (see, **?**] for a comprehensive coverage of stochastic programming). In this approach, the first stage consists of finding a set of a priori routes while the demands are not known yet with certainty. Once stochasticity reveals itself, the second stage consists of planning/obtaining a set of recourse decisions in the execution of each a priori route. The a priori optimization approach is a particularly suitable paradigm to model the VRPSD when the aim is to execute a route repeatedly over a long horizon. In the reoptimization approach, after the demand of each customer has been observed and served, the remaining portion of the vehicle route is conceptually reoptimized-by choosing the first customer to visit next and by deciding if a visit to the depot to replenish vehicle capacity should be performed first; see Secomandi [48] and Secomandi and Margot [49] for applications in which route reoptimization is allowed.

As mentioned before, under the a priori optimization approach for the VRPSD, a set of planned routes is determined in the first stage based on probabilistic information. To tackle the second-stage, a *recourse policy* must be designed. Such a policy corresponds to a set of predetermined rules to derive recourse decisions based on the residual capacity of the vehicle as well as the visits that are scheduled along the route. A recourse policy then provides the driver with a full *prescription* to react to incoming situations. Several recourse policies have been proposed. In the classical recourse policy, the driver follows the planned route until the vehicle capacity is depleted. Whenever the demand of a specific customer exceeds the residual capacity of the vehicle, the vehicle must execute a back-forth (BF) trip to the depot to replenish the capacity in order to complete the service. If the observed demand turns out to be equal to the residual capacity, the vehicle performs a restocking trip to the depot and then continues to the next customer. This classical policy was introduced by Dror and Trudeau [16] and implemented by Gendreau et al. [20], Hjorring and Holt [27], Laporte et al. [35], Rei et al. [44] and Jabali et al. [28]. As an alternative, one could apply an *optimal restocking policy* in which, the driver also prescribes preventive return (PR) trips to the depot in anticipation of potential failures whenever the residual capacity falls below a threshold value. In the optimal restocking policy, the vehicle prescribes PR trips in addition to BF trips such that the total expected cost is minimized, thus obtaining optimal customer-specific thresholds. This policy was introduced by Yee and Golden [58] and implemented by Yang et al. [57] and Bianchi et al. [7]. One also can consider rule based policies introduced by Salavati-Khoshghalb et al. [45], in which customer-specific thresholds are established in accordance with various operational rules. Salavati-Khoshghalb et al. [46] proposed a hybrid recourse policy, which combines two operational measures in order to prescribe PR trips.

To tackle the VRPSD modeled under the a priori paradigm, several exact, heuristic, and metaheuristic algorithms have been proposed; see for more details Gendreau et al. [22]. Two exact solution techniques have been used in this context. The Integer *L*-shaped algorithm and the column generation approach. The Integer *L*-shaped algorithm was adapted for the VRPSD by Gendreau et al. [20], Hjorring and Holt [27], Laporte et al. [35], and Jabali et al. [28]. The column generation approach was applied to the VRPSD by Christiansen and Lysgaard [13], as well as by Gauvin et al. [19]. All of these papers implemented the classical recourse policy. More recently, Salavati-Khoshghalb et al. [45] and Salavati-Khoshghalb et al. [46] have extended the Integer *L*-shaped algorithm to consider PR trips for rule-based policies. However, there are few research studies devoted to present and examine the optimal restocking policy. Yee and Golden [58] defined the optimal restocking recourse strategy, under which a set of optimal threshold-based recourse decisions including BF and PR trips can be obtained for given planned routes. Such an optimal restocking policy has been integrated in heuristic and meta-heuristic solution procedures to solve the VRPSD by Yang et al. [57] and Bianchi et al. [7]. Generally, these heuristic procedures result in overall sub-optimal pair of routing and recourse decisions.

Recently, Louveaux and Salazar-González [37] have integrated the optimal restocking policy in the a priori optimization solution approach to model the VRPSD. They propose an implementation of the *L*-shaped method to solve exactly the resulting problem. It should be noted that, while this paper provides bounding procedures applicable to instances in which customer demand distributions are not identical, much of the work focuses on the case where all customers have identical demand distributions and all their computational results cover only this case.

The purpose of this paper is to propose an exact algorithm to solve the VRPSD under an optimal restocking recourse policy, thus yielding solutions that are optimal both with respect to routing decisions and restocking ones. The proposed algorithm is an adaptation of the *L*-shaped method that uses various bound improvement procedures to achieve an effective performance. Furthermore, our approach allows for the consideration of different demand distributions for the customers in a computationally effective way, as long as they are discrete and with finite support, as shown by the numerical results that we report.

The remainder of this paper is organized as follows. Section §5.2 lays out the VRPSD model under the a priori approach with an optimal restocking policy. We devote Section §5.3 to propose an exact method, for solving the VRPSD under an optimal restocking policy, enhanced by various lower bounding schemes. Section §5.4 presents the results of a computational study to examine the performance of the proposed exact method. Section §5.5 proposes some conclusions and future research directions.

5.2 Optimal Restocking Recourse Policy Under the A Priori Approach

In Section §5.2.1, we first present the Vehicle Routing Problem with Stochastic Demands (VRPSD) modeled under the a priori optimization approach. To model the recourse problem, we recall the optimal restocking policy resulting in a set of optimal recourse decisions in §5.2.2.

5.2.1 VRPSD Formulation Under an A Priori Approach

This section revisits the VRPSD formulation presented by Gendreau et al. [20] and Laporte et al. [35]. 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 each having capacity Q is initially located. Each vertex v_i ($i = 2, \ldots, n$) represents a customer whose stochastic demand ξ_i follows a discrete probability distribution with a finite support, defined as the ordered set $\{\xi_i^1, \xi_i^2, \ldots, \xi_i^l, \ldots, \xi_i^{s^i}\}$, where $\xi_i^{s^i} \leq Q$. We denote by p_i^l, the probability of observing the l^{th} demand level, i.e., $\mathbb{P}[\xi_i = \xi_i^l] = p_i^l$. The traveling cost along an arc $(v_i, v_j) \in \mathcal{E}$ is denoted by c_{ij}, where the cost matrix $C = (c_{ij})$ is symmetric and satisfies the triangle inequality.

To formulate the VRPSD, we first recall the *a priori* optimization approach by Bertsimas et al. [4]. As previously mentioned, the first stage consists of making classical VRP routing decisions with probabilistic information about the stochastic demands. The decision variable x_{ij} ($i < j$) denotes the number of times edge (v_i, v_j) is traversed in the first-stage.

Given the notation devised previously in Gendreau et al. [20] and Laporte et al. [35], the a priori model for the VRPSD is formulated as follows:

$$\begin{aligned}
\underset{x}{\text{minimize}} \quad & \sum_{i<j} c_{ij} x_{ij} + \mathcal{Q}(x) && && (5.1)\\
\text{subject to} \quad & \sum_{j=2}^{n} x_{1j} = 2m, && && (5.2)\\
& \sum_{i<k} x_{ik} + \sum_{k<j} x_{kj} = 2, && k = 2, \dots, n && (5.3)\\
& \sum_{v_i, v_j \in S} x_{ij} \leq |S| - \left\lceil \frac{\sum_{v_i \in S} \mathbb{E}(\xi_i)}{Q} \right\rceil, && (S \subset \mathcal{V} \setminus \{v_1\}; 2 \leq |S| \leq n-2) && \\
& && && (5.4)\\
& 0 \leq x_{ij} \leq 1, && 2 \leq i < j < n && (5.5)\\
& 0 \leq x_{1j} \leq 2, && j = 2, \dots, n && (5.6)\\
& x = (x_{ij}), && \text{integer} && (5.7)
\end{aligned}$$

In this formulation, constraints (5.2) ensure that exactly m vehicle routes that start and end at the depot are established; constraints (5.3) ensure that each customer is connected to two other vertices; constraints (5.4) stand simultaneously as subtour elimination constraints and capacity constraints, which remove both subtours, and infeasible routes with an excessive expected demand. Then, the first-stage traveling costs are incurred in the objective function (5.1) as $\sum_{i<j} c_{ij} x_{ij}$.

Let us now suppose that an a priori routing solution x in model (5.1)-(5.7) is given. In the presence of demand stochasticity, however, an a priori route may fail at a specific customer at which the observed demand exceeds the residual capacity of the vehicle. Then, a recourse or corrective decision must be taken to either regain (i.e., in a reactive fashion) or preserve (i.e., in a proactive fashion) routing feasibility. In the context of the VRPSD, the recourse decisions are in the form of return trips to depot, but these trips entail extra costs. Then, the expected cost of the recourse actions that are taken given the routing solution x under a given policy is represented by $\mathcal{Q}(x)$ in the objective function (5.1).

Dror and Trudeau [16] have shown that, for route-based recourse policies, $\mathcal{Q}(x)$ can be decomposed by route. They also showed that the expected cost of recourse actions for a route depends on its orientation, i.e., in which direction it is executed. Thus, the expected recourse cost for routing solution x can be computed as (5.19), where $\mathcal{Q}^{r,\delta}$ denotes the expected recourse cost of the r^{th} a priori route in the orientation $\delta = 1, 2$.

$$\mathcal{Q}(x) = \sum_{r=1}^{m} \min\{\mathcal{Q}^{r,1}, \mathcal{Q}^{r,2}\}. \tag{5.8}$$

Computing $\mathcal{Q}^{r,\delta}$ for $\delta = 1, 2$ under an optimal restocking policy, thus obtaining a set of optimal recourse decisions for the r^{th} a priori route, is the subject of the next subsection.

5.2.2 The Optimal Restocking Policy

In this section we recall the optimal restocking policy, devised by Yee and Golden [58] for the VRPSD. Let us first consider an a priori route expressed as vector $\vec{v} = (v_1 = v_{i_1}, v_{i_2}, \ldots, v_{i_t}, v_{i_{t+1}} = v_1)$. An optimal restocking policy is a sequential decision rule that determines whether the vehicle after serving a specific customer with an arbitrary residual capacity onboard proceeds according to the planned route or performs a PR trip first. More precisely, let us assume that after serving the ${i_j}^{\text{th}}$ customer of the route, the residual capacity of the vehicle is equal to q units. If the vehicle proceeds to the following customer (i.e., i_{j+1}), then it must attempt to satisfy the stochastic demand $\xi_{i_{j+1}}$. When $q \geq \xi_{i_{j+1}}$ service is completed with a nonnegative residual capacity of $q - \xi_{i_{j+1}}$, and one must again decide whether the vehicle should proceed or replenish the vehicle capacity first. If $q < \xi_{i_{j+1}}$, then a route failure occurs and the vehicle must perform a BF trip (at the cost of $2c_{1,i_{j+1}}$) before completing the service of customer i_{j+1} with a residual capacity equal to $Q + q - \xi_{i_{j+1}}$. It should be noted that we also consider a fixed cost b for each route failure as Yang et al. [57]; this penalizes the disruption at a customer location caused by the second vehicle visit. On the other hand, the vehicle can replenish its capacity by performing a PR trip in order to avoid potential route failures, before starting the service at the ${i_{j+1}}^{\text{th}}$ customer. After replenishing the vehicle capacity at the cost of $c_{1,i_j} + c_{1,i_{j+1}} - c_{i_j,i_{j+1}}$, the vehicle can fulfill all demand observations of customer

i_{j+1} since $Q \geq \xi_{i_{j+1}}$, and then will decide whether to serve the following customer i_{j+2} with a residual capacity equal to $Q - \xi_{i_{j+1}}$, or perform a PR trip.

Let $F_{i_j}(q)$ be the optimal onward recourse cost-to-go after serving the $i_j{}^{\text{th}}$, and remaining with a residual capacity of q. Then, the optimal expected recourse cost of the a priori route $\vec{v}$ can be expressed by using the following Bellman equation,

$$F_{i_j}(q) = \min \begin{cases} H_{i_j,i_{j+1}}(q) : \sum\limits_{k:\xi^k_{i_{j+1}} \leq q} F_{i_{j+1}}(q - \xi^k_{i_{j+1}}) p^k_{i_{j+1}} + \\ \qquad\qquad \sum\limits_{k:\xi^k_{i_{j+1}} > q} [b + 2c_{1,i_{j+1}} + F_{i_{j+1}}(Q + q - \xi^k_{i_{j+1}})] p^k_{i_{j+1}}, \\ H'_{i_j,i_{j+1}}(q) : c_{1,i_j} + c_{1,i_{j+1}} - c_{i_j,i_{j+1}} + \sum\limits_{k=1}^{s_i} F_{i_{j+1}}(Q - \xi^k_{i_{j+1}}) p^k_{i_{j+1}} \end{cases} \tag{5.9}$$

where, $H_{i_j,i_{j+1}}(q)$ and $H'_{i_j,i_{j+1}}(q)$ express the total costs associated to the proceeding and restocking decisions after serving the $i_j{}^{\text{th}}$ customer, respectively. This computation differs from the formula given by Yang et al. [57], since it only considers the recourse cost and not the total cost of the route. Using equation (5.9), we have $F_{i_{t+1}}(.) = 0$ since after serving the last customer the expected recourse cost is equal to zero. We note that $F_{i_j}(q)$ is an optimal policy only if $F_{i_{j+1}}(.), F_{i_{j+2}}(.), \ldots, F_{i_t}(.)$ are already optimally given. Furthermore, let $\vec{\theta}^* = (\theta^*_{i_1}, \theta^*_{i_2}, \ldots, \theta^*_{i_j}, \ldots, \theta^*_{i_t})$ be the optimal restocking policy threshold vector. Since $F_{i_j}(q)$ is monotonically non-increasing with respect to q, $\theta^*_{i_j} = min\{q | H_{i_j,i_{j+1}}(q) \leq H'_{i_j,i_{j+1}}(q)\}$ (for further details see, e.g., Yee and Golden [58] and Yang et al. [57]). Based on $\theta^*_{i_j}$ computed by the latter equation, the optimal decision at the $i_j{}^{\text{th}}$ customer is either replenishing the vehicle capacity for $q < \theta^*_{i_j}$ or proceeding to the next customer whenever $q \geq \theta^*_{i_j}$.

Given equation (5.9) and assuming that the r^{th} vehicle performs the a priori route, its expected recourse cost can then be computed for the first orientation (i.e., $\delta = 1$) as follows,

$$\mathcal{Q}^{r,1} = F_{i_1}(Q). \tag{5.10}$$

To compute the expected recourse cost of the route for the second orientation (i.e., $\mathcal{Q}^{r,2}$), we reapply function (5.10) to the reverse of the a priori route $\vec{v}$.

5.3 An Integer L-shaped Algorithm to Solve the VRPSD under an Optimal Restocking Policy

In this section, we adapt the Integer *L*-shaped algorithm to exactly solve the VRPSD under an optimal restocking recourse policy. The Integer *L*-shaped algorithm is proposed by Laporte and Louveaux [34] to tackle two-stage stochastic integer program with recourse. It stands as a general branch-and-cut (B&C) procedure in which, feasibility cuts and branching are employed to obtain integer first-stage solutions. A feasible integer solution with an excessive expected recourse cost is removed by adding optimality cuts. The optimality cuts which are originally developed by Laporte and Louveaux [34], adjust a lower bound for $\mathcal{Q}(x)$ at each feasible integer solution using its combinatorial structure locally. However, the Integer *L*-shaped algorithm solely relying on optimality cuts may turn to an implicit enumeration procedure of feasible integer solutions. Therefore, there is a need to provide lower bounding procedures enhancing the B&C procedure.

Such lower bound improving procedures were first proposed by Hjorring and Holt [27] (for the VRPSD with classical recourse) via the concept of *partial routes*, which are feasible fractional solutions with certain structures. These new valid inequalities called lower bounding functional (LBF) cuts improve lower bounds for several integer feasible solutions. However, some restrictive assumptions are made: 1) all customers demands are discrete, independent and uniformly distributed and 2) a maximum of one failure can occur within the fractional structure. The concept of partial routes was then developed by Laporte et al. [35] for multi-VRPSD, where customer demands follow continuous distributions. Jabali et al. [28] generalize the concept of partial routes proposed by Hjorring and Holt [27] through defining various structures, thus improving global lower bound for many fractional feasible solutions.

In this section we apply LBF cuts of Jabali et al. [28] to the case of optimal restocking policy when customers demand are defined through arbitrary discrete distributions. The LBF cuts of Jabali et al. [28] are only applied to the case where customer demands are

Normal distributions. To do so, we provide several approximation schemes to compute valid lower bounds for the expected recourse cost of partial routes under an optimal restocking policy. In subsection §5.3.1, we first revisit the Integer L-shaped algorithm. Then, in subsection §5.3.2 we present a lower bounding scheme to approximate $\mathcal{Q}(x)$, where x contains partial routes of Jabali et al. [28]. In subsection §5.3.3, we provide a general lower bound L where $L \leq \mathcal{Q}(x)$ and x satisfies (5.2)-(5.7).

5.3.1 The Integer L-Shaped Algorithm

In this section we describe the Integer L-shaped employed to optimally solve the VRPSD in a general B&C procedure. In this B&C procedure a master problem, called *current problem* (CP) is established by relaxing capacity and subtour elimination constraints as well as the integrality requirements. The expected recourse function $\mathcal{Q}(x)$ is replaced by the continuous variable Θ and is initially bounded from below by a general lower bound L using (5.14). The first current problem CP^0 can be presented by (5.11), (5.2), (5.3),(5.5), (5.6), and (5.14). At an arbitrary iteration ν, CP^ν is shown in the following model,

$$CP^\nu : \min_{x,\Theta} \quad \sum_{i<j} c_{ij} x_{ij} + \Theta \tag{5.11}$$

subject to (5.2), (5.3), (5.5), (5.6),

$$\sum_{v_i, v_j \in S^k} x_{ij} \leq |S^k| - \left\lceil \frac{\sum_{v_i \in S^k} \mathbb{E}(\xi_i)}{Q} \right\rceil \quad \forall k \in \mathbf{ST}^{\nu-1}, S^k \subset \mathcal{V} \setminus \{v_1\}, 2 \leq |S^k| \leq n-2, \tag{5.12}$$

$$L + (\Theta_p^q - L)\left(\sum_{h \in \mathbf{PR}^q} W_p^h(x) - |\mathbf{PR}^q| + 1 \right) \leq \Theta \ \forall q \in \mathbf{PS}^{\nu-1}, p \in \{\alpha, \beta, \gamma\}, \tag{5.13}$$

$$L \leq \Theta \tag{5.14}$$

$$\sum_{\substack{1 \leq i \leq j \\ x_{ij}^f = 1}} x_{ij} \leq \sum_{1 \leq i \leq j} x_{ij}^f - 1 \qquad \forall f \in \mathbf{OC}^{\nu-1}, \tag{5.15}$$

where, constraints (5.12), (5.13), and (5.15) respectively are subtour elimination and capacity constraints, LBF cuts, and optimality cuts. At each iteration ν, an optimal solution (x^{ν}, Θ^{ν}) is obtained by solving CP^{ν}. Violated capacity and subtour elimination constraints (5.12) are added to CP^{ν} until no more violated cuts are detected. We denote by $\{k'\}$ the index set associated to the subsets of vertices violating (5.12) at iteration ν. We also denote by $\mathbf{ST}^{\nu-1}$ the set of index sets of the vertices violating (5.12) in the first $\nu - 1$ iterations. Then, at iteration ν we set $\mathbf{ST}^{\nu} = \mathbf{ST}^{\nu-1} \cup \{k'\}$. The separation procedure is performed by the CVRP package of [38]. When no violated constraint (5.12) is detected, the lower bounding cuts (5.13) are added to strength the overall bounding scheme. An exact separation procedure developed by Jabali et al. [28] detects partial solutions within x^{ν}. We denote by $\{q'\}$ the index set associated to partial solutions identified in iteration ν. We also denote by $\mathbf{PS}^{\nu-1}$ the set of index sets of the partial solutions detected to add (5.13) in the first $\nu - 1$ iterations. Then, at iteration ν we set $\mathbf{PS}^{\nu} = \mathbf{PS}^{\nu-1} \cup \{q'\}$. Each partial solution contains a set of partial routes, here at iteration ν denoted by h' including various structures α, β, and γ proposed by Jabali et al. [28]. The expected recourse cost associated to each structure $p \in \{\alpha, \beta, \gamma\}$ is computed as $\Theta_p^{q'}$ using the procedure presented in subsection §5.3.2. We also denote by $\mathbf{PR}^{\nu-1}$ the set of partial routes detected in the first $\nu - 1$ iterations. Then, at iteration ν we set $\mathbf{PR}^{\nu} = \mathbf{PR}^{\nu-1} \cup \{h'\}$. The branching scheme obtains integrality requirements whenever needed. At integer feasible solutions, $\mathcal{Q}(x^{\nu})$ is computed to update the upper bound,. In the case of $\Theta^{\nu} < \mathcal{Q}(x^{\nu})$, an optimality cut (5.15) is added to CP^{ν}. We denote by $\{f'\}$ the index set of x^{ν} when an optimality cut is added. We also denote by $\mathbf{OC}^{\nu-1}$ the set of index sets of vertices associated to the optimality cuts detected in the first $\nu - 1$ iterations. Then, at iteration ν we set $\mathbf{OC}^{\nu} = \mathbf{OC}^{\nu-1} \cup \{f'\}$.

5.3.2 Approximating an Optimal Restocking Policy

Here, we present the bounding procedures to approximate the expected recourse cost of partial solutions. At an arbitrary iteration ν, we assume that partial solutions within x^{ν} are detected, here denoted by q, using the exact procedure proposed by Jabali et al. [28]. We note that Θ_p^q in (5.13) is set as the sum of the lower bounds of the various partial routes (or routes) included in q and can be computed by $\Theta_p^q = \sum\limits_{h \in \mathbf{PR}^q} \Theta_p^{qh}$. We then drop the index q in Θ_p^{qh} and present it by Θ_p^h. In this section, we describe an approximation technique to compute Θ_p^h in order to add LBF cuts (5.13). In (5.13), Θ_p^h presents a valid lower bound for the expected recourse cost of partial route h with an arbitrary structure $p \in \{\alpha, \beta, \gamma\}$. In what follows, we only derive Θ_α^ν. The approximating technique can then be applied to compute Θ_β^h and Θ_γ^h because β and γ topologies can be viewed as successions of the α topology.

Let $h \in \mathbf{PR}^\nu$ be a partial route with the α topology. Generally, a partial route consists of an alternation of *chains* and *unstructured components*. The vertices of a chain are connected in the support graph at iteration ν (denoted by $\bar{\mathcal{G}}^\nu$); $x_{ij}^\nu = 1$ in $\bar{\mathcal{G}}^\nu$ then there is an edge (v_i, v_j). The vertex set of a chain is called chain vertex set (CVS). The vertex set of each unstructured components is called unstructured vertex set (UVS). Each UVS lies between two chains and connected to them at articulation vertices. Partial route h with α topology consists of two chains $S_h^1 = \{v_{h_1}^1, \ldots, v_{|S_h^1|}^1\}$ and $S_h^2 = \{v_{h_1}^2, \ldots, v_{|S_h^2|}^2\}$ and one unstructured set U_h^1 as $h = (v_1 = v_{h_1}^1, \ldots, v_{|S_h^1|}^1, U_h^1, v_{h_1}^2, \ldots, v_{|S_h^2|}^2 = v_1)$, where $U_h^1 = \{v_{u_1}, v_{u_2}, \ldots, v_{u_l}\}$; $v_{|S_h^1|}^1$ and $v_{h_1}^2$ are articulation vertices which connect chains S_h^1 and S_h^2 to U_h^1, respectively.

For the sake of simplicity, we redefine the partial route h, in similar terms as a route, as follows

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

where the articulation vertices $v_{|S_h^1|}^1$ and $v_{h_1}^2$ are denoted by $v_{i_{j-l}}$ and $v_{i_{j+1}}$, respectively. We define an artificial route $\tilde{h}$ associated to the partial route h as follows,

$$\tilde{h} = (v_1 = v_{i_1}, \ldots, v_{i_{j-l}}, \square_{i_{j-l+1}}, \square_{i_{j-l+2}}, \ldots, \square_{i_j}, v_{i_{j+1}}, \ldots, v_{i_{t+1}} = v_1), \qquad (5.16)$$

where each ordering of l unsequenced customers 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 ${i_j}^{\text{th}}$ position in the artificial route $\tilde{h}$. Then, we develop a bounding procedure for the artificial route $\tilde{h}$.

Approximation:

To compute a valid lower bound for the expected recourse cost, we need to provide some additional notations. Let $s = (i_a, q)$ denote the state of the system (i.e., the vehicle) after serving the i_ath customer of the a priori route $\vec{v} = (v_1 = v_{i_1}, v_{i_2}, \ldots, v_{i_{j-l}}, \ldots, v_{i_a}, v_{i_{a+1}}, \ldots, v_{i_{j+1}}, \ldots, v_{i_t}, v_{i_{t+1}} = v_1)$ with q units of the residual capacity onboard, as in the Bellman equation (5.9). When performing the a priori route $\vec{v}$ (or more generally for two successive customers in a chain), the system will make a transition from state $s = (i_a, q)$ to some state $s' = (i_{a+1}, q')$. Furthermore, one can easily determine all possible values of q' and use them to compute $F_{i_a}(q)$. When dealing with artificial route $\tilde{h}$, things are not as easy, since the customers between $v_{i_{j-l}}$ and $v_{i_{j+1}}$ are not known exactly. In that portion of the artificial route, we must associate *pseudo states* which are associated not with specific customers, but rather to *positions in the route*. Thus, we let $\mathsf{s} = (\square_{i_a}, q)$ represent the state of the system after serving the (still unknown) customer in the i_ath position of the artificial route.

In the following, we present a successive approximation scheme that computes a valid lower bound for the optimal cost-to-go value function for pseudo state s, denoted by $\widetilde{F}_{i_a}(\mathsf{s} = (\square_{i_a}, q))$. Based on the Bellman's principle of optimality, we also suppose that the optimal (or, a valid lower bound) cost-to-go value function $\widetilde{F}_{i_{a+1}}(\mathsf{s}' = (\square_{i_{a+1}}, q'))$ has been determined beforehand, for all $s' = (\square_{i_{a+1}}, q')$. Let us now define the auxiliary value $\hat{F}_{i_a}(\mathsf{s} = (\square_{i_a}, q), \mathsf{s}' = (v_{u_1}, q'))$, which corresponds to a *conditional lower bound* on the optimal cost-to-go value function, if we assume that customer $v_{u_1} \in U_h^1$ occupies the i_{a+1}th position (i.e., $\square_{i_{a+1}} := v_{u_1}$ in s'). We can then write

$$\hat{F}_{i_a}(\mathsf{s} = (\square_{i_a}, q), \mathsf{s}' = (v_{u_1}, q')) =$$

$$= \min \begin{cases} \displaystyle\sum_{k:\xi^k_{u_1} \leq q} \widetilde{F}_{i_{a+1}}(\mathsf{s}' = (v_{u_1}, q' := q - \xi^k_{u_1}))p^k_{u_1} + \\ \displaystyle\sum_{k:\xi^k_{u_1} > q} [b + 2c_{1,u_1} + \widetilde{F}_{i_{a+1}}(\mathsf{s}' = (v_{u_1}, q' := Q + q - \xi^k_{u_1}))]p^k_{u_1}, \\ c_{1,i_a} + c_{1,u_1} - c_{i_k,u_1} + \displaystyle\sum_{k=1}^{s_{u_1}} \widetilde{F}_{i_{a+1}}(\mathsf{s}' = (v_{u_1}, q' := Q - \xi^k_{u_1}))p^k_{u_1}. \end{cases} \tag{5.17}$$

To compute $\hat{F}_{i_a}(\mathsf{s} = (\square_{i_k}, q), \mathsf{s}' = (v_{u_1}, q'))$ in (5.17), the PR trip travel cost is replaced by a lower bound $\underset{v_{u_e} \in U^1_h : v_{u_e} \neq v_{u_1}}{minimum} \{c_{1,u_e} + c_{1,u_1} - c_{u_e,u_1}\}$. To determine an *unconditional lower bound* on $\widetilde{F}_{i_a}(\mathsf{s} = (\square_{i_a}, q))$, we simply take the minimum of the conditional lower bounds, i.e., we set

$$\widetilde{F}_{i_a}(\mathsf{s} = (\square_{i_a}, q)) = \underset{v_{u_e} \in U^1_h}{min} \hat{F}_{i_a}(\mathsf{s} = (\square_{i_a}, q), \mathsf{s}' = (v_{u_e}, q')). \tag{5.18}$$

There are two boundary cases which differ from the situation presented above. The first case arises when we start the approximation scheme, where $\mathsf{s} = (\square_{i_j}, q)$ and $s' = (v_{i_{j+1}}, q')$. In this case, we can compute directly the unconditional lower bound on the optimal cost-to-go value function. The PR trip cost can be obtained by $\underset{v_{u_e} \in U^1_h}{minimum}\{c_{1,u_e} + c_{1,i_{j+1}} - c_{u_e,i_{j+1}}\}$. The second case arises in the last step of overall scheme, where $s = (v_{i_{j-l}}, q)$ and $\mathsf{s}' = (\square_{i_{j-l+1}}, q')$. In this case, the PR trip costs for each v_{u_1} in $\hat{F}_{i_{j-l}}(s = (v_{i_{j-l}}, q), \mathsf{s}' = (\square_{i_{j-l+1}} := v_{u_1}, q'))$ can be computed as $c_{1,u_1} + c_{1,i_{j-l}} - c_{i_{j-l},u_1}$. The latter boundary case will result in an unconditional bound $\widetilde{F}_{i_{j-l}}(s = (i_{j-l}, q))$.

It should be noted that the the optimal cost-to-go functions $F_{i_{j+1}}(.), F_{i_{j+2}}(.), \ldots, F_{i_t}(.)$ can be exactly computed by the Bellman equation (5.9). Then, the bounding procedure described above provides an unconditional lower bound on $\widetilde{F}_{i_{j-l}}(s = (i_{j-l}, q))$ $\forall q$. Next, the unconditional lower bound $\widetilde{F}_{i_{j-l}}(s = (i_{j-l}, q))$ can be applied in (5.9) to successively compute unconditional lower bounds $\widetilde{F}_{i_{j-l-1}}(.), \widetilde{F}_{i_{j-l-2}}(.), \ldots, \widetilde{F}_{i_1}(.)$. We set $\widetilde{F}_{i_1}(Q)$ as the valid lower bound for the expected recourse cost of artificial route $\tilde{h}$

in the first direction and denote it by $\widetilde{F}^1_{i_1}(Q)$. By reversing $\tilde{h}$ and applying the bounding procedure we will obtain a valid lower bound for the second direction, denoted by $\widetilde{F}^2_{i_1}(Q)$. We then set

$$\Theta^h_\alpha = \min\{\widetilde{F}^1_{i_1}(Q), \widetilde{F}^2_{i_1}(Q)\} \tag{5.19}$$

where, Θ^h_α is a valid lower bound for the expected recourse cost of partial route h, detected in the partial solutions q within optimal first-stage solution x^ν at iteration ν. Moreover, we note that partial routes with β and γ topologies consist of several partial routes with α topology and we can apply the same procedure to compute Θ^h_β and Θ^h_γ. Finally, we set $\Theta^q_p = \sum\limits_{h \in \mathbf{PR}^q} \Theta^h_p$ for $p \in \{\alpha, \beta, \gamma\}$ to be used in LBF cuts (5.13).

5.3.3 General Lower Bound

In this subsection, we propose a procedure to obtain a general lower bound L to be used in constraints (5.13) and (5.14). As defined by Laporte and Louveaux [34], the expected recourse cost associated to the feasible solution x^L with minimum expected recourse cost corresponds to a general lower bound. Laporte and Louveaux [33] were the first authors to present a general lower bound for the VRPSD under the classical recourse. The quality of the general lower bound presented in Laporte and Louveaux [33] is further improved by Laporte et al. [35]. Suppose that $\vec{v}^1$, $\vec{v}^2$, $\ldots$, $\vec{v}^m$ are the vehicle routes contained in x^L. Using the notation of Laporte and Louveaux [34],

$$L = \mathcal{Q}(x^L) \leq \min_x\{\mathcal{Q}(x) | (5.2) - (5.6)\} = \sum_{k=1}^m \min\{\mathcal{Q}^{k,1}(\vec{v}^k), \mathcal{Q}^{k,2}(\vec{v}^k)\}. \tag{5.20}$$

For computing L in (5.20), we assume that: the vehicle route denoted by $\vec{v}^{12}$ is obtained by concatenating $\vec{v}^2$ after $\vec{v}^1$; v_{l1} and v_{f2} present the last customer in $\vec{v}^1$, and the first customer in $\vec{v}^2$, respectively; $F^{\vec{v}^1}_{v_1}(Q)$ and $F^{\vec{v}^2}_{v_1}(Q)$ are the expected recourse costs associated to $\vec{v}^1$ and $\vec{v}^2$, respectively; $\bar{F}^{\vec{v}^{12}}_{v_{l1}}(.)$ and $F^{\vec{v}^{12}}_{v_{l1}}(.)$ are the expected recourse costs from the depot to v_{l1} and expected cost-to-go from v_{l1} to the depot going through $\vec{v}^2$, respectively; and $p^q_{v_{l1}}$ is the probability of having q units of residual capacity after serving customer

v_{l1}.

The expected recourse cost of $\vec{v}^{12}$ in the first direction can be computed as follows,

$$F^{\vec{v}^{12}}_{v_1}(Q) = \sum_q \{\bar{F}^{\vec{v}^{12}}_{v_{l1}}(q) + F^{\vec{v}^{12}}_{v_{l1}}(q)\} p^q_{v_{l1}}. \tag{5.21}$$

By definition, we have

$$F^{\vec{v}^{12}}_{v_{l1}}(q) = \min \begin{cases} \displaystyle\sum_{k:\xi^k_{v_{f2}} \leq q} F^{\vec{v}^{12}}_{v_{f2}}(q - \xi^k_{v_{f2}}) p^k_{v_{f2}} + \\ \displaystyle\sum_{k:\xi^k_{v_{f2}} > q} [b + 2c_{1,v_{f2}} + F^{\vec{v}^{12}}_{v_{f2}}(Q + q - \xi^k_{v_{f2}})] p^k_{v_{f2}}, \\ c_{1,v_{l1}} + c_{1,v_{f2}} - c_{v_{l1},v_{f2}} + \displaystyle\sum_{k=1}^{s_{v_{f2}}} F^{\vec{v}^{12}}_{v_{f2}}(Q - \xi^k_{v_{f2}}) p^k_{v_{f2}}. \end{cases} \tag{5.22}$$

We also have $F^{\vec{v}^{12}}_{v_{l1}}(q) \leq c_{1,v_{l1}} + c_{1,v_{f2}} - c_{v_{l1},v_{f2}} + \sum_{k=1}^{s_{v_{f2}}} F^{\vec{v}^{12}}_{v_{f2}}(Q - \xi^k_{v_{f2}}) p^k_{v_{f2}}$ which coupled with (5.21) results in

$$F^{\vec{v}^{12}}_{v_1}(Q) \leq \sum_q \{\bar{F}^{\vec{v}^{12}}_{v_{l1}}(q) + c_{1,v_{l1}} + c_{1,v_{f2}} - c_{v_{l1},v_{f2}} + \sum_{k=1}^{s_{v_{f2}}} F^{\vec{v}^{12}}_{v_{f2}}(Q - \xi^k_{v_{f2}}) p^k_{v_{f2}}\} p^q_{v_{l1}}. \tag{5.23}$$

Assuming that $\vec{v}^{12}$ is equivalent to the concatenation of $\vec{v}^1$ and $\vec{v}^2$, the relation (5.23) can further yield

$$F^{\vec{v}^{12}}_{v_1}(Q) \leq c_{1,v_{l1}} + c_{1,v_{f2}} - c_{v_{l1}} + F^{\vec{v}^1}_{v_1}(Q) + F^{\vec{v}^2}_{v_1}(Q),$$

where, the first term in (5.23) is equivalent to $F^{\vec{v}^1}_{v_1}(Q)$ in the backward fashion and the last term in (5.23) is equivalent to $F^{\vec{v}^2}_{v_1}(Q)$ in the forward fashion.

We perform the same procedure to concatenate the remaining routes $\vec{v}^3, \ldots, \vec{v}^m$ to

$\vec{v}^{12}$ and conclude that:

$$F_{v_1}^{\vec{v}^{1\ldots m}}(Q) \leq \sum_{k=1}^{m-1} c_{\mathrm{PR}}^{k} + \sum_{k=1}^{m} F_{v_1}^{\vec{v}^{k}}(Q) \tag{5.24}$$

where $\vec{v}^{1\ldots m}$ is obtained by the successive concatenation of all routes and c_{PR}^{k} denotes the k^{th} least PR trip cost.

The desired L can be obtained by bounding $\sum_{k=1}^{m} F_{v_1}^{\vec{v}^{k}}(Q)$. However, the vehicle routes $\vec{v}^{1}$, $\vec{v}^{2}$, …, $\vec{v}^{m}$, as well as $\vec{v}^{1\ldots m}$ are not known, but we can use the fact that the route $\vec{v}^{1\ldots m}$ in the left-hand-side of (5.24) consists of all customers. To calculate a general lower bound $L^* \leq L$, we can approximate the left-hand-side of (5.24) by constructing a large unstructured set $U_L = \mathcal{V} \setminus \{v_1\}$. Then, one can reduce the problem of finding a valid lower bound for U_L to computing the minimum expected recourse cost $\widetilde{F}_{v_1}^{\tilde{l}_z}(Q)$ of artificial routes $\tilde{l}_z$ for $z = 2, \ldots, n$, which are obtained by only fixing the last customer before returning to the depot v_z, i.e.,

$$\tilde{l}_z = (v_1 = v_{i_1}, \square_{i_2}, \square_{i_3}, \ldots, \square_{i_{t-1}}, v_z, v_{i_{t+1}} = v_1). \tag{5.25}$$

This is done exactly as in §5.3.2. Finally, a general lower bound L^* can be computed as

$$L^* = \min_{z:2,\ldots,n} \widetilde{F}_{v_1}^{\tilde{l}_z}(Q) - \sum_{k=1}^{m-1} c_{\mathrm{PR}}^{k}. \tag{5.26}$$

5.4 Numerical Results

In this section, we evaluate the quality of the proposed Integer L-shaped algorithm by conducting computational experiments of instances. Overall, we present the numerical result for three sets of instances.

Symmetric Instances: In the first set of instances (which is made up of the instances of Salavati-Khoshghalb et al. [45]), customer locations and demands are randomly generated. We generated instances consisting of a set of n vertices as $\{v_1, \ldots, v_n\}$, in which v_1 represents the depot and $n - 1$ customers and all vertices are randomly scattered in

$[0, 100]^2$ according to a continuous uniform distribution. In the first set, each customer is randomly (i.e., with equal probability) assigned to one of the three demand ranges $[1, 5]$, $[6, 10]$, $[11, 15]$ and then five realizations in each range are observed accordingly to the probabilities $\{0.1, 0.2, 0.4, 0.2, 0.1\}$.

Asymmetric Instances: In the second set of instances, customer locations are the same as symmetric instances. Each customer is randomly (i.e., with equal probability) assigned to one of the five demand ranges $[1,5]$, $[6, 10]$, $[11, 15]$, $[4, 7]$, and $[9, 12]$. Each of the first three demand ranges has five possible demand values, the occurrence of each which (in ascending order) is expressed with the following probabilities $\{0.1, 0.2, 0.4, 0.2, 0.1\}$. Each of the last two demand ranges has four possible demand values, the occurrence of each which (in ascending order) is expressed with the following probabilities $\{0.4, 0.3, 0.2, 0.1\}$.

In what follows, all settings are considered in both symmetric and asymmetric instances. The traveling cost c_{ij} is set as the Euclidean distance between each pair v_i and v_j and rounded to the nearest integer. The filling coefficient $\bar{f}$ is equal to $\frac{\sum_{i=2}^{n} \mathbb{E}(\xi_i)}{mQ}$. Four filling coefficients $\bar{f} = 0.90, 0.92, 0.94,$ and 0.96 are considered. The capacity of each vehicle is directly inferred from $\bar{f}$. We consider 11 combinations of (n, m) for each of the four filling coefficients, as detailed in Table 5.I. We generated 10 instances for each entry of the table. Thus, our generated test bed contains 440 instances, overall 880 runs for symmetric and asymmetric instances.

Table 5.I – 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

In our computational result, a fixed cost denoted by $b = \sum\limits_{i=2,\ldots,n} c_{i1}/(n-1)$ is incurred when experiencing route failures. We recall that b primarily penalizes disruption at a customer location caused by the second vehicle visit.

The Integer L-shaped algorithm and the bounding scheme are coded in C++ using ILOG CPLEX 12.6. The subtour elimination and capacity constraints (5.4) are identified using the CVRPSEP package of Lysgaard et al. [38]. The general branch-and-cut framework as the Integer L-shaped algorithm is implemented using the OOBB package developed by Gendron et al. [24]. Computational experiments were conducted on a cluster of 27 machines, each having two Intel(R) Xeon(R) X5675 3.07 GHz processors with 12 cores and 96 GB of RAM running Linux. An integer feasible solution with a relative optimality gap less than 0.01% is assumed optimal. Also, a maximum CPU run time of 10 hours is imposed on all runs. If the maximum allotted time is reached, we then report the best integer solution obtained.

The Instances Generated by Louveaux and Salazar-González [37]: The instances of Louveaux and Salazar-González [37] are selected from benchmark instances `E031-09h`, `E051-05e`, `E076-07s`, and `E101-08e`, see `http://neo.lcc.uma.es/vrp/vrp-instances/`. However, the expected demand of all customers is set to $\mu = 5$. Parameter K denotes the number of possible demand realizations for each customer, for each instance a single value of K is applied to all customers. Namely, $K = 3$ or $K = 9$. Then, for all $j \in V \setminus \{v_1\}$ and $k = 1, \ldots, K$, stochastic demands are generated by $\xi_j^k = \mu - \lfloor K/2 \rfloor + k - 1$. The probability of each demand realization ξ_j^k is then computed by $p_j^k = k/\lceil K/2 \rceil^2$ for $k < \lceil K/2 \rceil^2$ and $p_j^k = (K - k + 1)/\lceil K/2 \rceil^2$ otherwise. The number of vehicle denoted by m is set to 2 and 3. The vehicle capacity is obtained by $Q = \max\{\lceil (n\mu)/(m\bar{f}) \rceil; \lceil n/m \rceil \mu\}$ in which the filling rates $\bar{f} = 0.90, 0.95$ are considered for $m = 2$ and in the case of $m = 3$ the filling rates $\bar{f} = 0.85, 0.90$. Also, Louveaux and Salazar-González [37] a fixed cost of $\Delta = 0, 10, 100$ as loading/unloading cost is considered for both BF and PR trips. In our recourse function, we denote by b a fixed cost as the customer unsatisfactory in the failure events.

In subsection 5.4.1, the performance of the Integer L-shaped algorithm as an exact solution method is evaluated in terms of various quality measures. We further compare the results of our optimal restocking policy with by pricing the optimal solutions under the classical policy. In subsection 5.4.2, based on the identical instances generated by Louveaux and Salazar-González [37] a separate set of experiments was made to compare the performance of the Integer *L*-shaped algorithm implemented in this paper.

5.4.1 Quality of the Integer L-Shaped Algorithm: Symmetric and Asymmetric Instances

We now present the computational result, expressing the performance of the proposed exact algorithm in Tables 5.II and 5.IV for symmetric and asymmetric instances. The conducted experiments are aggregated according to the pair (n,m) and the filling coefficient $\bar{f}$. Tables 5.II and 5.IV contains the following notations: 1) the "Solved" columns present the number of instances (out of ten for each aggregated category) that were solved to optimality by the algorithm; 2) columns present the number of instances (out of ten for each aggregated category) that were solved with an optimality gap $\leq 1\%$; 3) the "Run(sec)" columns refer to the average running times in seconds that were needed by the algorithm to solve those instances to optimality; 4) 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).

By analyzing the computational results in Tables 5.II and 5.IV, we observe similar trends that were reported by Gendreau et al. [20], Laporte et al. [35], and Jabali et al. [28] for the classical recourse policy. These trends indicate that an increase in the filling rate and/or the number of vehicles results in a reduction of the optimally solved instances, an increase in the running time to solve instances optimally, and an increase in the optimality gap, which all together present an increase in the overall complexity of the VRPSD instances. Moreover, when compared to the filling rate, the number of vehicles seems to have a more substantial impact on the complexity of the instances. As reported in Tables 5.II and 5.IV, the Integer *L*-shaped algorithm implemented in this paper optimally solves 227 and 242 each out of 440 runs from symmetric and asymmetric instances, respectively; these correspond to 51.6% and 55.0% of the generated instances. The overall average optimality gaps are 0.83% and 0.80%, respectively. Moreover, the proposed algorithm solves 285 and 297 runs with an optimality gap $\leq 1\%$ of symmetric

Table 5.II – Performance measures for symmetric instances.

n	m	$\bar{f}$	solved	$\leq 1\%$	Run(sec)	Gap	$\bar{f}$	solved	$\leq 1\%$	Run(sec)	Gap	$\bar{f}$	solved	$\leq 1\%$	Run(sec)	Gap	$\bar{f}$	solved	$\leq 1\%$	Run(sec)	Gap
20	2	0.90	10	10	13.90	0.00%	0.92	10	10	10.00	0.00%	0.94	10	10	1.70	0.00%	0.96	10	10	60.60	0.00%
30	2	0.90	10	10	6.40	0.00%	0.92	8	10	1.12	0.05%	0.94	10	10	2134.30	0.00%	0.96	9	9	1261.89	0.18%
40	2	0.90	10	10	15.60	0.00%	0.92	10	10	80.90	0.00%	0.94	10	10	8.20	0.00%	0.96	9	10	683.22	0.00%
40	3	0.90	5	9	713.20	0.30%	0.92	8	8	4991.38	0.39%	0.94	4	8	11504.25	0.81%	0.96	2	3	19371.50	1.17%
40	4	0.90	0	2	0.00	2.45%	0.92	0	1	0.00	3.82%	0.94	0	1	0.00	3.01%	0.96	0	1	0.00	4.21%
50	2	0.90	10	10	20.60	0.00%	0.92	9	9	1412.78	0.19%	0.94	10	10	44.20	0.00%	0.96	7	10	457.57	0.16%
50	3	0.90	4	7	1609.50	1.03%	0.92	4	8	8656.25	0.67%	0.94	3	8	1199.67	0.78%	0.96	1	4	1262.00	1.38%
50	4	0.90	2	2	331.00	3.68%	0.92	1	1	7775.00	3.26%	0.94	0	2	0.00	2.50%	0.96	0	0	0.00	3.61%
60	2	0.90	10	10	1017.00	0.00%	0.92	9	10	82.78	0.02%	0.94	8	10	574.25	0.07%	0.96	7	10	1580.86	0.11%
60	3	0.90	3	7	2978.33	0.77%	0.92	1	4	757.00	2.14%	0.94	1	3	1006.00	1.91%	0.96	1	2	32411.00	2.02%
60	4	0.90	0	2	0.00	2.82%	0.92	0	1	0.00	2.86%	0.94	0	2	0.00	3.28%	0.96	1	1	9785.00	4.06%
Average					474.00	0.74%				1624.43	0.89%				1376.79	0.82%				2437.91	1.13%
Total			64	79				60	72				56	74				47	60		

Table 5.III – Average savings vs classical recourse with respect to total cost (Sav1), and recourse cost (Sav2) for the first set.

n	m	$\bar{f}$	Sav1	Sav2	$\bar{f}$	Sav1	Sav2	$\bar{f}$	Sav1	Sav2	$\bar{f}$	Sav1	Sav2
20	2	0.90	0.42%	40.57%	0.92	0.86%	45.16%	0.94	1.78%	53.47%	0.96	2.61%	55.38%
30	2	0.90	0.33%	31.21%	0.92	0.26%	40.78%	0.94	0.71%	44.39%	0.96	1.39%	53.85%
40	2	0.90	0.08%	43.11%	0.92	0.31%	47.34%	0.94	0.48%	55.08%	0.96	1.22%	62.90%
40	3	0.90	0.16%	36.69%	0.92	0.30%	35.98%	0.94	0.68%	38.94%	0.96	1.67%	57.43%
40	4	0.90	0.00%	0.00%	0.92	0.00%	0.00%	0.94	0.00%	0.00%	0.96	0.00%	0.00%
50	2	0.90	0.03%	49.58%	0.92	0.27%	56.73%	0.94	0.68%	59.56%	0.96	0.95%	64.37%
50	3	0.90	0.09%	46.08%	0.92	0.10%	43.37%	0.94	0.35%	53.32%	0.96	2.33%	64.37%
50	4	0.90	0.12%	26.61%	0.92	0.12%	23.61%	0.94	0.00%	0.00%	0.96	0.00%	0.00%
60	2	0.90	0.14%	54.11%	0.92	0.19%	54.05%	0.94	0.39%	55.01%	0.96	0.97%	67.57%
60	3	0.90	0.12%	30.33%	0.92	0.26%	76.15%	0.94	0.12%	36.01%	0.96	1.21%	56.21%
60	4	0.90	0.00%	0.00%	0.92	0.00%	0.00%	0.94	0.00%	0.00%	0.96	2.24%	39.80%
Average			0.18%	42.15%		0.35%	46.82%		0.77%	52.09%		1.53%	59.65%

Table 5.IV – Performance measures for asymmetric instances.

n	m	$\bar{f}$	solved	$\leq 1\%$	Run(sec)	Gap	$\bar{f}$	solved	$\leq 1\%$	Run(sec)	Gap	$\bar{f}$	solved	$\leq 1\%$	Run(sec)	Gap	$\bar{f}$	solved	$\leq 1\%$	Run(sec)	Gap
20	2	0.90	10	10	51.90	0.00%	0.92	10	10	74.80	0.00%	0.94	10	10	0.30	0.00%	0.96	10	10	848.40	0.00%
30	2	0.90	10	10	4.30	0.00%	0.92	10	10	103.30	0.00%	0.94	10	10	365.40	0.00%	0.96	10	10	976.90	0.00%
40	2	0.90	10	10	8.10	0.00%	0.92	10	10	40.30	0.00%	0.94	9	10	184.89	0.02%	0.96	10	10	658.80	0.00%
40	3	0.90	5	9	2285.40	0.23%	0.92	7	8	4587.86	0.46%	0.94	4	9	3594.00	0.28%	0.96	2	6	21227.00	1.01%
40	4	0.90	1	3	2207.00	1.93%	0.92	1	2	238.00	2.71%	0.94	0	2	0.00	3.11%	0.96	0	1	0.00	2.89%
50	2	0.90	10	10	68.10	0.00%	0.92	10	10	318.60	0.00%	0.94	9	10	18.22	0.00%	0.96	10	10	4675.00	0.00%
50	3	0.90	7	8	2883.14	0.79%	0.92	4	5	4153.75	1.04%	0.94	6	8	6655.67	0.27%	0.96	1	3	1720.00	1.27%
50	4	0.90	2	2	9245.50	3.13%	0.92	1	2	1478.00	2.49%	0.94	0	1	0.00	3.28%	0.96	0	0	0.00	3.79%
60	2	0.90	8	10	511.00	0.10%	0.92	9	10	728.22	0.05%	0.94	8	10	1693.88	0.05%	0.96	9	10	2866.00	0.09%
60	3	0.90	5	8	6968.40	0.21%	0.92	0	6	0.00	1.19%	0.94	2	5	11888.00	0.96%	0.96	1	4	6095.00	1.43%
60	4	0.90	1	1	11020.00	4.77%	0.92	0	1	0.00	3.26%	0.94	0	3	0.00	3.35%	0.96	0	0	0.00	3.59%
Average					1501.17	0.74%				1005.97	0.75%				1674.52	0.76%				2785.92	0.94%
Total			69	81				62	74				58	78				53	64		

Table 5.V – Average savings vs classical recourse with respect to total cost (Sav1), and recourse cost (Sav2) for the second set.

n	m	$\bar{f}$	Sav1	Sav2	$\bar{f}$	Sav1	Sav2	$\bar{f}$	Sav1	Sav2	$\bar{f}$	Sav1	Sav2
20	2	0.90	0.27%	27.97%	0.92	0.86%	42.68%	0.94	1.07%	45.48%	0.96	3.10%	57.52%
30	2	0.90	0.33%	38.92%	0.92	0.20%	39.72%	0.94	0.46%	43.24%	0.96	1.58%	52.68%
40	2	0.90	0.12%	39.13%	0.92	0.07%	48.11%	0.94	0.36%	59.29%	0.96	1.18%	64.97%
40	3	0.90	0.13%	46.04%	0.92	0.40%	40.22%	0.94	0.61%	43.66%	0.96	2.10%	48.42%
40	4	0.90	0.53%	31.11%	0.92	0.37%	35.28%	0.94	0.00%	0.00%	0.96	0.00%	0.00%
50	2	0.90	0.16%	49.76%	0.92	0.24%	56.14%	0.94	0.20%	56.19%	0.96	0.80%	58.47%
50	3	0.90	0.09%	35.75%	0.92	0.87%	58.17%	0.94	0.83%	57.10%	0.96	1.82%	56.66%
50	4	0.90	0.16%	24.06%	0.92	0.43%	43.94%	0.94	0.00%	0.00%	0.96	0.00%	0.00%
60	2	0.90	0.07%	48.68%	0.92	0.24%	56.14%	0.94	0.27%	50.97%	0.96	0.83%	66.41%
60	3	0.90	0.17%	37.89%	0.92	0.00%	0.00%	0.94	0.56%	41.98%	0.96	1.13%	62.16%
60	4	0.90	0.13%	39.10%	0.92	0.00%	0.00%	0.94	0.00%	0.00%	0.96	0.00%	0.00%
Average			0.18%	39.64%		0.37%	47.83%		0.54%	50.61%		1.53%	59.43%

and asymmetric instances, respectively.

In order to qualify the magnitude of savings obtained by performing the optimal restocking policy, we execute the optimal solutions under the classical recourse. Tables 5.III and 5.V illustrate the comparisons of two recourse policies with respect to the total cost denoted by “Sav1”$= \frac{Q^{class.}(x)-Q^{opt}(x)}{cx+Q^{class.}(x)} \times 100$ and recourse cost as “Sav2”$= \frac{Q^{class.}(x)-Q^{opt}(x)}{Q^{class.}(x)} \times 100$. The weighted average savings in terms of “Sav1” are 0.65% and 0.61% for symmetric and assymetric instances, respectively. In terms of “Sav2”, the weighted average savings are 49.46% and 48.70%, respectively.

Also, in order to qualify the magnitude of savings obtained by performing the optimal restocking policy we compare the total cost of the optimal solutions obtained under optimal restocking policy with optimal solutions under both the best rule-based policy presented by Salavati-Khoshghalb et al. [45] and the best hybrid policy recourse presented by in Salavati-Khoshghalb et al. [46].

Tables 5.VI and 5.VII express the latter comparisons with respect to the total cost as “Sav3”$= \frac{Q^{rule.}(x^*_{rule})-Q^{opt}(x^*_{opt})}{cx^*_{rule}+Q^{rule}(x^*_{rule})} \times 100$ and “Sav4”$= \frac{Q^{hybrid}(x^*_{hybrid})-Q^{opt}(x^*_{opt})}{cx^*_{hybrid}+Q^{hybrid}(x^*_{hybrid})} \times 100$, respectively. In Sav3 and Sav4, x^*_{opt}, x^*_{rule}, and x^*_{hybrid} are optimal routing decisions obtained by solving the VRPSD instances under optimal restocking policy, best rule-based and hybrid recourse policies, respectively. As presented in Tables 5.VI and 5.VII, the best rule-based policy shows less deviation from optimal restocking policy.

Table 5.VI – Average savings vs rule-based recourse $\eta\bar{\xi}_{i_{j+1}}$ for $\eta = 1$, with respect to total cost.

n	m	$\bar{f}$	Sav3	$\bar{f}$	Sav3	$\bar{f}$	Sav3	$\bar{f}$	Sav3
20	2	0.90	0.056%	0.92	0.034%	0.94	0.083%	0.96	0.153%
30	2	0.90	0.015%	0.92	0.007%	0.94	0.042%	0.96	0.100%
40	2	0.90	0.004%	0.92	0.005%	0.94	0.033%	0.96	0.088%
40	3	0.90	0.016%	0.92	0.009%	0.94	0.018%	0.96	0.068%
40	4	0.90	0.000%	0.92	0.000%	0.94	0.000%	0.96	0.000%
50	2	0.90	0.006%	0.92	0.011%	0.94	0.019%	0.96	0.075%
50	3	0.90	0.010%	0.92	0.011%	0.94	0.015%	0.96	0.089%
50	4	0.90	0.000%	0.92	0.006%	0.94	0.000%	0.96	0.000%
60	2	0.90	0.007%	0.92	0.011%	0.94	0.015%	0.96	0.057%
60	3	0.90	0.001%	0.92	0.028%	0.94	0.001%	0.96	0.033%
60	4	0.90	0.000%	0.92	0.000%	0.94	0.000%	0.96	0.000%
Average			0.015%		0.013%		0.034%		0.096%

Table 5.VII – Average savings vs hybrid recourse policy for $\{\underline{\theta},\bar{\theta}\}$=$\{0.35, 0.65\}$, with respect to total cost.

n	m	$\bar{f}$	Sav4	$\bar{f}$	Sav4	$\bar{f}$	Sav4	$\bar{f}$	Sav4
20	2	0.90	0.119%	0.92	0.165%	0.94	0.809%	0.96	1.259%
30	2	0.90	0.041%	0.92	0.007%	0.94	0.153%	0.96	3.076%
40	2	0.90	0.004%	0.92	0.141%	0.94	0.499%	0.96	0.397%
40	3	0.90	0.016%	0.92	0.076%	0.94	0.501%	0.96	0.954%
40	4	0.90	0.000%	0.92	0.000%	0.94	0.000%	0.96	0.000%
50	2	0.90	0.032%	0.92	0.074%	0.94	0.296%	0.96	0.854%
50	3	0.90	0.010%	0.92	0.011%	0.94	0.734%	0.96	0.741%
50	4	0.90	0.052%	0.92	0.006%	0.94	0.000%	0.96	0.000%
60	2	0.90	0.027%	0.92	0.057%	0.94	0.030%	0.96	0.679%
60	3	0.90	0.001%	0.92	0.028%	0.94	0.001%	0.96	0.000%
60	4	0.90	0.000%	0.92	0.000%	0.94	0.000%	0.96	0.000%
Average			0.039%		0.086%		0.378%		1.296%

5.4.2 The instances Generated by Louveaux and Salazar-González [37]

We have compared the solutions that we obtain with those of Louveaux and Salazar-González [37] for the instances that both methods are able to solve. This comparison confirmed that our method provides valid results. Regarding computational times, Louveaux and Salazar-González's implementation seems to be more effective than ours: if one accounts for differences between the machine that they have used and ours, their code runs faster and it is able to solve to optimality more instances than our algorithm for a given CPU time allowance. This result is not surprising given the fact that their approach uses specialized procedures for instances with identical demand distributions, which is not the case of our method. Moreover, it is observed from Tables 5.VIII-5.X that LBF cuts developed in this paper can significantly reduce the number of branch-and-cut nodes explored by the Integer L-shaped algorithm. The number of B&C nodes explored in the proposed method in this paper is much smaller than Louveaux and Salazar-González's implementation, in which their B&C procedure turns to an enumeration.

5.5 Conclusions

In this paper, we develop an exact solution methodology to solve the VRPSD under an optimal restocking policy. To do so, the Integer L-shaped algorithm is adapted. To enhance the efficiency of the Integer L-shaped algorithm, various lower bounding schemes are redeveloped. The key element for successfully employing such bounding procedures is to provide effective lower approximation of the expected recourse cost of

Table 5.VIII – Louveaux and Salazar-González [37] with $\Delta = 0$.

	Instance							Our result						Louveaux and Salazar-González [37]			
Instance	Veh.	$\bar{f}$	Scen.	Node	Run(min)	Gap	Routing	Recourse	OptRestock	Classical	L	Node	Routing	Recourse	Gap	Run(min)	L
E031-09h	2	0.90	3	12	0.00	0.00	332	0.752989	332.752989	332.773660	0.000000	325	332	0.7530	0	0.00	0.0000
E031-09h	2	0.95	3	115	0.00	0.00	334	1.295551	335.295551	335.382649	0.000000	2035	334	1.2956	0	0.02	0.0000
E031-09h	2	0.90	9	2447	0.17	0.00	334	3.673848	337.673848	337.813496	0.000000	3632	334	3.6738	0	0.04	0.0000
E031-09h	2	0.95	9	11923	6.28	0.00	334	10.525120	344.525120	345.235935	0.000000	36654	334	10.5251	0	1.30	0.0000
E031-09h	3	0.85	3	305	0.03	0.00	358	0.947237	358.947237	358.996960	0.000000	17950	358	0.9472	0	0.28	0.0000
E031-09h	3	0.90	3	3303	0.90	0.00	364	0.065544	364.065544	364.066998	0.000000	94518	364	0.0655	0	4.07	0.0000
E031-09h	3	0.85	9	22825	21.08	0.00	361	6.155214	367.155214	367.343869	0.000000	248044	361	6.1552	0	18.17	0.0000
E031-09h	3	0.90	9	92237	5*h*.	0.64	361	11.783918	372.783918	373.184593	0.000000	604022	363	10.1294	2.325	5 h.	0.0000
E051-05e	2	0.90	3	13	0.00	0.00	441	0.000234	441.000234	441.000239	0.000000	3260	441	0.0002	0	0.06	0.0000
E051-05e	2	0.95	3	139	0.01	0.00	441	0.311264	441.311264	441.330315	0.000000	3889	441	0.3113	0	0.10	0.0000
E051-05e	2	0.90	9	3967	1.29	0.00	441	2.006072	443.006072	443.089529	0.000000	10709	441	2.0061	0	0.30	0.0000
E051-05e	2	0.95	9	51292	5*h*.	0.59	441	7.082580	448.082580	448.501497	0.000000	314798	441	7.0826	0.130	5 h.	0.0000
E051-05e	3	0.85	3	17	0.01	0.00	459	0.002388	459.002388	459.002388	0.000000	6557	459	0.0000	0	0.16	0.0000
E051-05e	3	0.90	3	7	0.01	0.00	459	0.049098	459.049098	459.049896	0.000000	3449	459	0.0491	0	0.07	0.0000
E051-05e	3	0.85	9	1489	2.44	0.00	459	1.550320	460.550320	460.572987	0.000000	22525	459	1.5503	0	0.62	0.0000
E051-05e	3	0.90	9	80279	5*h*.	0.21	460	5.629006	465.629006	465.727164	0.000000	303297	460	5.6290	0	72.44	0.0000
E076-07s	2	0.90	3	1	0.00	0.00	549	0.004528	549.005528	549.005529	0.000000	757	549	0.0045	0	0.03	0.0000
E076-07s	2	0.95	3	187	0.06	0.00	550	0.163882	550.163882	550.165034	0.000000	7869	550	0.1639	0	0.34	0.0000
E076-07s	2	0.90	9	897	0.27	0.00	550	0.815011	550.815011	550.841649	0.000000	8522	550	0.8150	0	0.27	0.0000
E076-07s	2	0.95	9	47267	5*h*.	0.35	550	4.799920	554.799920	555.006649	0.000000	425613	550	4.7999	0	252.76	0.0000
E076-07s	3	0.85	3	251	0.65	0.00	567	0.125934	567.125934	567.134489	0.000000	43343	567	0.1259	0	2.75	0.0000
E076-07s	3	0.90	3	4033	27.59	0.00	568	1.266042	569.266042	569.374859	0.000000	213546	568	1.2660	0	36.14	0.0000
E076-07s	3	0.85	9	22870	5*h*.	0.14	568	1.952144	569.952144	570.001057	0.000000	440721	568	1.9521	0	107.42	0.0000
E076-07s	3	0.90	9	20039	5*h*.	0.50	570	3.249967	573.249967	573.328432	0.000000	579000	571	3.3077	1.029	5 h.	0.0000
E101-08e	2	0.90	3	1	0.00	0.00	640	0.001000	640.001000	640.001003	0.000000	2819	640	0.0010	0	0.19	0.0000
E101-08e	2	0.95	3	46525	5*h*.	0.07	640	1.731616	641.731616	641.808065	0.000000	83765	640	1.7316	0	29.32	0.0000
E101-08e	2	0.90	9	44929	169.55	0.00	640	1.304019	641.304019	641.453231	0.000000	34436	640	1.3040	0	13.05	0.0000
E101-08e	2	0.95	9	26237	5*h*.	0.83	640	6.119062	646.119062	646.959141	0.000000	172619	640	6.1191	0.851	5 h.	0.0000
E101-08e	3	0.85	3	2036	9.40	0.00	655	0.354229	655.354229	655.355447	0.000000	9025	655	0.3542	0	0.98	0.0000
E101-08e	3	0.90	3	26418	5*h*.	0.05	657	1.296878	658.296878	658.388005	0.000000	40442	657	1.2969	0	9.07	0.0000
E101-08e	3	0.85	9	8483	5*h*.	0.59	655	4.483187	659.483187	659.527152	0.000000	292928	657	1.9841	0	105.42	0.0000
E101-08e	3	0.90	9	8172	5*h*.	1.36	657	9.554738	666.554738	666.886476	0.000000	267310	658	9.5547	1.545	5 h.	0.0000

Table 5.IX – Louveaux and Salazar-González [37] with $\Delta = 10$.

Instance	Instance			Our result								Louveaux and Salazar-González [37]					
Instance	Veh.	$\bar{f}$	Scen.	Node	Run(min)	Gap	Routing	Recourse	OptRestock	Classical	L	Node	Routing	Recourse	Gap	Run(min)	L
E031-09h	2	0.90	3	34	0.00	0.00	332	1.303910	333.303910	333.850296	0.000912	501	332	1.3039	0	0.01	0.0029
E031-09h	2	0.95	3	99	0.00	0.00	334	2.282922	336.282922	337.387534	1.080402	1586	334	2.2829	0	0.02	0.8886
E031-09h	2	0.90	9	2757	0.15	0.00	334	5.919955	339.919955	340.550842	1.600999	3952	334	5.9200	0	0.04	1.9888
E031-09h	2	0.95	9	24169	12.71	0.00	334	16.259292	350.259292	351.763999	4.648780	59413	334	16.2593	0	3.29	5.0661
E031-09h	3	0.85	3	333	0.03	0.00	358	1.211586	359.211586	359.666162	0.000000	21184	358	1.2116	0	0.26	0.0000
E031-09h	3	0.90	3	3255	0.66	0.00	364	0.104780	364.104780	364.244267	0.017250	74042	364	0.1048	0	2.99	0.0348
E031-09h	3	0.85	9	46755	30.55	0.00	364	5.732663	369.732663	370.519772	0.865468	400531	364	5.7327	0	79.79	1.8020
E031-09h	3	0.90	9	97781	5*h*.	1.18	364	13.607132	377.607132	379.140990	2.810705	594758	363	16.1500	3.169	5 h.	4.1332
E051-05e	2	0.90	3	13	0.00	0.00	441	0.000472	441.000472	441.001141	0.000005	2938	441	0.0005	0	0.05	0.0002
E051-05e	2	0.95	3	219	0.03	0.00	441	0.639656	441.639656	441.979402	0.159510	3067	441	0.6397	0	0.06	0.2956
E051-05e	2	0.90	9	12391	5.22	0.00	441	3.491032	444.491032	444.863469	0.612693	13347	441	3.4910	0	0.40	1.3256
E051-05e	2	0.95	9	51150	5*h*.	1.35	441	11.658757	452.658757	453.668452	2.813372	371415	441	11.6588	0.494	5 h.	4.0884
E051-05e	3	0.85	3	17	0.00	0.00	459	0.008354	459.008354	459.022057	0.000000	7475	459	0.0000	0	0.19	0.0000
E051-05e	3	0.90	3	13	0.00	0.00	459	0.078449	459.078449	459.140321	0.000062	6298	459	0.0784	0	0.14	0.0264
E051-05e	3	0.85	9	523	0.33	0.00	459	2.409384	461.409384	461.689472	0.061817	12581	459	2.4094	0	0.43	0.7701
E051-05e	3	0.90	9	48015	5*h*.	0.89	459	12.800608	471.800608	472.856052	0.921362	392822	460	9.1329	0	106.49	3.1460
E076-07s	2	0.90	3	1	0.00	0.00	549	0.009033	549.010033	549.015684	0.000000	829	549	0.0090	0	0.03	0.0000
E076-07s	2	0.95	3	350	0.16	0.00	550	0.423725	550.423725	550.605126	0.022705	6334	550	0.4237	0	0.28	0.2339
E076-07s	2	0.90	9	4437	2.54	0.00	550	1.603081	551.603081	551.775532	0.197213	8455	550	1.6031	0	0.29	0.7090
E076-07s	2	0.95	9	47144	5*h*.	0.89	550	8.595511	558.595511	559.248449	2.081761	91136	550	8.5955	0	20.43	3.4147
E076-07s	3	0.85	3	1370	2.04	0.00	568	0.001891	568.001891	568.003728	0.000000	47440	567	0.2371	0	2.92	0.0000
E076-07s	3	0.90	3	12730	5*h*.	0.24	570	1.794271	571.794271	572.397456	0.000000	353364	570	0.0626	0	58.78	0.0003
E076-07s	3	0.85	9	19500	5*h*.	0.35	570	2.247197	572.247197	572.461363	0.005996	673940	571	0.9262	0.523	5 h.	0.2483
E076-07s	3	0.90	9	17778	5*h*.	0.92	571	5.213997	576.213997	576.693552	0.420190	559695	571	5.2140	1.066	5 h.	1.9884
E101-08e	2	0.90	3	1	0.00	0.00	640	0.002235	640.002235	640.003675	0.000000	2885	640	0.0022	0	0.23	0.0000
E101-08e	2	0.95	3	30172	5*h*.	0.36	643	0.548675	643.548675	643.715440	0.002874	217505	643	0.5487	0.329	5 h.	0.0317
E101-08e	2	0.90	9	52052	5*h*.	0.14	640	2.333638	642.333638	642.612766	0.074250	110106	640	2.3336	0	44.34	0.3934
E101-08e	2	0.95	9	15481	5*h*.	1.36	640	10.294426	650.294426	651.450881	1.464995	163776	640	10.2944	1.083	5 h.	2.7507
E101-08e	3	0.85	3	269	1.93	0.00	655	0.777460	655.778460	656.033373	0.000000	10229	655	0.7775	0	0.75	0.0000
E101-08e	3	0.90	3	3217	21.58	0.00	657	1.921698	658.921698	659.361185	0.000000	90335	657	1.9217	0	17.52	0.0002
E101-08e	3	0.85	9	10898	5*h*.	0.41	655	7.295404	662.296862	662.635669	0.000458	471952	657	2.8075	0.162	5 h.	0.1218
E101-08e	3	0.90	9	11672	5*h*.	1.49	657	13.327256	670.327256	671.049059	0.115445	239545	668	4.0220	2.039	5 h.	1.2579

Table 5.X – Louveaux and Salazar-González [37] with $\Delta = 100$.

	Instance			Our result								Louveaux and Salazar-González [37]					
Instance	Veh.	$\bar{f}$	Scen.	Node	Run(min)	Gap	Routing	Recourse	OptRestock	Classical	L	Node	Routing	Recourse	Gap	Run(min)	L
E031-09h	2	0.90	3	169	0.01	0.00	334	0.036972	334.036972	334.147639	0.003998	669	334	0.0370	0	0.01	0.0322
E031-09h	2	0.95	3	3265	0.37	0.00	334	11.169266	345.169266	355.431492	5.190493	1129	334	11.1693	0	0.01	9.7750
E031-09h	2	0.90	9	59511	5*h*.	3.04	334	25.855218	359.855218	365.186955	8.598236	4989	334	25.8552	0	0.06	21.8767
E031-09h	2	0.95	9	74436	5*h*.	9.75	334	67.120808	401.120808	410.516574	30.996017	46716	334	67.1208	0	2.18	55.7272
E031-09h	3	0.85	3	1471	0.23	0.00	358	3.573217	361.573217	365.688988	0.000002	52467	358	3.5732	0	1.63	0.0003
E031-09h	3	0.90	3	3821	1.13	0.00	364	0.452652	364.452652	365.839687	0.050483	101055	364	0.4527	0	6.67	0.3827
E031-09h	3	0.85	9	127177	5*h*.	3.74	364	23.823043	387.823043	394.813032	3.143872	541383	364	23.8230	0	158.13	19.8215
E031-09h	3	0.90	9	78789	5*h*.	9.58	366	54.263840	420.263840	432.489493	12.249430	575050	364	55.1384	2.570	5 h.	45.4655
E051-05e	2	0.90	3	17	0.00	0.00	441	0.002620	441.002620	441.009260	0.000026	2733	441	0.0026	0	0.06	0.0024
E051-05e	2	0.95	3	14766	25.34	0.00	441	3.595178	444.595178	447.821180	0.712359	2852	441	3.5952	0	0.07	3.2511
E051-05e	2	0.90	9	46241	5*h*.	2.29	441	16.760808	457.760808	460.828923	3.383876	14095	441	16.7608	0	0.43	14.5815
E051-05e	2	0.95	9	26849	5*h*.	7.07	442	51.976878	493.976878	500.614472	18.781663	368184	441	52.5861	0.530	5 h.	44.9723
E051-05e	3	0.85	3	20	0.01	0.00	459	0.062053	459.062053	459.199085	0.000000	8771	459	0.0000	0	0.32	0.0000
E051-05e	3	0.90	3	107	0.03	0.00	459	0.342615	459.342615	459.954143	0.000271	8710	459	0.3426	0	0.22	0.2906
E051-05e	3	0.85	9	49377	5*h*.	1.66	465	9.691321	474.691321	477.304968	0.232376	20158	459	10.1148	0	0.97	8.4706
E051-05e	3	0.90	9	27171	5*h*.	7.54	465	40.333816	505.333816	512.781498	3.920572	375607	460	40.6002	0	117.98	34.6055
E076-07s	2	0.90	3	1	0.00	0.00	549	0.049578	549.050578	549.107086	0.000000	1433	549	0.0496	0	0.06	0.0002
E076-07s	2	0.95	3	25005	197.20	0.00	550	2.762314	552.762314	554.565950	0.129071	4344	550	2.7623	0	0.17	2.5724
E076-07s	2	0.90	9	32914	5*h*.	0.96	550	8.695714	558.695714	560.180479	1.162683	17355	550	8.6957	0	0.91	7.7989
E076-07s	2	0.95	9	20332	5*h*.	5.28	551	43.766938	594.766938	599.337753	14.337217	2079	557	43.2132	0	0.09	37.5621
E076-07s	3	0.85	3	1417	3.89	0.00	567	1.238007	568.238007	569.336837	0.000000	90188	568	0.0098	0	9.19	0.0000
E076-07s	3	0.90	3	15759	5*h*.	0.69	574	0.434985	574.434985	574.928142	0.000000	472268	570	0.4141	0	189.78	0.0035
E076-07s	3	0.85	9	23601	5*h*.	1.37	573	5.347527	578.347527	579.545348	0.029557	583500	571	3.0801	0.496	5 h.	2.7316
E076-07s	3	0.90	9	15355	5*h*.	5.61	574	31.493586	605.493586	610.692594	2.006243	313769	579	25.1061	2.687	5 h.	21.8722
E101-08e	2	0.90	3	1	0.00	0.00	640	0.013355	640.013355	640.027718	0.000000	1296	640	0.0134	0	0.10	0.0000
E101-08e	2	0.95	3	16934	5*h*.	0.66	645	0.384135	645.384135	645.696159	0.018200	176070	645	0.3841	0.657	5 h.	0.3482
E101-08e	2	0.90	9	20995	5*h*.	1.32	643	6.560891	649.560891	650.559998	0.481734	184880	643	6.5609	0.654	5 h.	4.3275
E101-08e	2	0.95	9	13665	5*h*.	4.65	644	38.262388	682.262388	686.048197	10.511748	186210	645	35.4935	1.467	5 h.	30.2575
E101-08e	3	0.85	3	16055	5*h*.	0.01	655	4.586534	659.587534	662.125702	0.000000	101256	657	0.0016	0	45.68	0.0000
E101-08e	3	0.90	3	18444	5*h*.	0.55	663	0.340071	663.340071	663.644677	0.000000	334797	661	0.6416	0.333	5 h.	0.0022
E101-08e	3	0.85	9	13544	5*h*.	2.33	664	11.504967	675.504967	677.378463	0.002695	217234	659	16.5437	2.873	5 h.	1.3399
E101-08e	3	0.90	9	13786	5*h*.	5.85	660	41.696028	701.696028	705.597480	0.630127	145040	682	16.3244	4.211	5 h.	13.8368

partial routes. In addition, a general lower bound enhancing the Integer *L*-shaped algorithm is developed. Using the exact method proposed in this paper, we are able to optimally solve problems with up to 60 customers and a fleet of four vehicles. It should be noted that the proposed exact method for the first time is able to solve the VRPSD instances in which customer demands can follow arbitrary discrete distributions.

The numerical results conducted in this paper show that the resulting routes from the optimal restocking policy yield a reasonable amount of savings, when compared to executing the classical policy on the same routes.

The present paper provides a standard procedure to bound the recourse costs associated with the optimal recourse policy in the context of the VRPSD. The bounding procedure proposed in this paper can be further applied to approximate optimal policies in different contexts like stochastic inventory routing problem which opens different research avenues.

CHAPTER 6

CONCLUSION

Vehicle routing problem with stochastic demands refers to pickup/delivery routing problems at operational level of a transportation company. In this variant of the classical vehicle routing problem, customer demands are only known through probability distributions. In this setting, the actual demand of each customer is revealed upon arriving at the customer's location. Therefore, the vehicle, executing a planned route obtained by forecasted values associated to the probability distributions here the expected demand of customers, may fail to complete the service a specific customer at which the observed demand exceeds the residual capacity of the vehicle. This is called route failure. Then, corrective, i.e., recourse actions must be taken to regain routing feasibility. In the context of VRPSD, recourse actions are in the form of returning trips to the depot location in order to replenish the vehicle capacity. To regain routing feasibility where a route failure occurred, the vehicle can perform a back-and-forth trip to the depot, then completes the service and continues to the next customer. In an anticipation of future failures, the vehicle can execute a preventive return whenever the residual capacity after serving the current customer falls below a threshold value. To model the problem, a recourse policy which is a full prescription to execute a set of determined recourse actions must be designed.

The efficiency of a recourse policy depends on several criteria as stated in the following points; 1) flexibility to perform diverse recourse actions, 2) diversity in operational and governing rules to perform recourse actions, 3) simplicity to understand and execute for the drivers, 4) preserving customer satisfactory, etc. In its broadest picture, this dissertation considers the mentioned points to design efficient and practical recourse policies.

The first paper focused on designing recourse policies which implement proactive recourse actions in the static fashion. We particularly introduced the concept of a rule-based recourse policy for the VRPSD and provided its formulation. The rule-based policy is defined through the thresholds based on 1) the vehicle capacity, 2) the expected demand of next customer, and 3) the total expected demand of unvisited customers. We note that we were the first to develop an exact solution method (the integer L-shaped algorithm) to solve the VRPSD under the volume rule-based recourse policies. To enhance the efficiency of the proposed algorithm, we developed various bounding procedures to improve the global lower bounding schemes. Our numerical experiments showed that the proposed volume-based policies outperform the classical policy, in reducing the number of route failures and the total costs. The proposed solution approach was able to efficiently solve a wide range of problems, with varying size and different filling rates.

The second part of this dissertation devoted to introduce a mixed recourse policy, for the first time, in the context of the VRPSD. Such mixed recourse policy combines the risk of failure and the distances to travel to govern the execution of recourse actions. Using the proposed risk measure, the driver replenishes the vehicle capacity, before visiting the next customer, only if the risk of the route failure is more than a preset risk threshold. Also, the driver will proceed the planned route whenever the risk of failure at the next customer is too low. Otherwise, when the risk of failure is neither too high nor too low, we employ a distance measure. We redeveloped the integer L-shaped algorithm to solve the VRPSD under the mixed recourse policy. To assess the effectiveness of the proposed method as well as the quality of the mixed recourse policy, we conducted an extensive computational experiments. We observed that our solution method is able to solve problems with up to 60 customers and a fleet of four vehicles to optimality. Moreover, the result clearly showed that the mixed policy presented in the second paper reduces the expected number of route failures.

The third part of this dissertation examined the optimal restocking policy in the VRPSD context. We modeled the underlying multi-VRPSD under an optimal restocking policy. We then proposed an an exact algorithm to solve the VRPSD, thus resulting in solutions that are optimal with respect to routing and restocking decisions. A successive approximation scheme, employed in the L-shaped algorithm, is devised to approximate the expected recourse cost under an optimal restocking policy. Then, a valid general lower bound is established to further enhance the overall branch-and-cut procedure. The numerical experiments showed that solving the VRPSD under the optimal restocking policy provides significant improvements in the total cost comparing with the classical recourse.

The integer L-shaped algorithm as a general B&C procedure must employ bounding procedures to tighten the optimality gap. The optimality gap in VRPSD is inherently wider than CVRP because of expected recourse cost incurred in the objective function in the VRPSD. It should be noted that the expected recourse cost is only bounded by the general lower bound. Then, the integer L-shaped algorithm can turn to an enumeration procedures. Therefore, there is a need to provide various bounding procedures helping to efficiently exploit quality solutions by tightening the optimality gap. In general we developed several bounding procedures which enhance the integer L-shaped algorithm when tackling the VRPSD under various recourse policies. To our best knowledge, approximating the expected recourse cost when customer demands follow general discrete distributions and presenting a general lower bound to initially bound the recourse function is proposed for the first time. It should be noted that LBF cuts are the only constraints which enable us to tighten the optimality gap and avoid enumeration.

6.1 Future works and perspectives

We conclude this dissertation by drawing avenues for future researches. Future studies can be grouped in the three following directions:

1. As mentioned in Chapter 1, in the present research we aimed at designing efficient recourse policies to solve underling VRPSD instances exactly. Although the integer L-shaped algorithm redeveloped in this paper is able to optimally solve instances up to 60 customers being served by 4 vehicles and 100 customers with 3 vehicles, designing sophisticated heuristics and meta-heuristics solution framework to tackle larger problems is of interest. It should be noted that few research works are provided to solve the multi-VRPSD using heuristics and meta-heuristics solution techniques.

2. Overall, three recourse policies are examined in this research associated to the

VRPSD context. Using various recourse policies examined in the current research, further work can aim at designing efficient recourse policies for several stochastic optimization problems like the VRP with stochastic travel and service times, VRP with stochastic demands and customers, multi-compartment VRPSD, stochastic inventory problems, etc.

3. In this dissertation, various bounding procedures by approximating (providing valid lower bounds) the expected recourse cost are proposed. The quality of LBF cuts profoundly depend on the mentioned valid lower bounds. Although by improving the quality of such valid lower bounds the quality of LBF cuts will be improved, the efficiency of overall B&C can be decreased. The further investigation can be focused on the tradeoff between the quality of LBF cuts and the efficiency of the integer L-shaped algorithm.
4. In this dissertation we developed specific approximation techniques to compute various valid lower bounds of the expected recourse cost associated to the solutions with certain structures. Such approximation techniques can be applied to approximate various recourse policies in different fields of research in order to provide valid lower bounds which speedup the tightening the optimality gap.

BIBLIOGRAPHY

[1] Aykagan Ak and Alan L. Erera. A paired-vehicle recourse strategy for the vehicle-routing problem with stochastic demands. *Transportation Science*, 41(2):222–237, 2007.

[2] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust optimization*. Princeton University Press, 2009.

[3] Jacques F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik*, 4(1):238–252, 1962.

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

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

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

[7] Leonora Bianchi, Mauro Birattari, Marco Chiarandini, Max Manfrin, Monaldo Mastrolilli, Luis Paquete, Olivia Rossi-Doria, and Tommaso Schiavinotto. Metaheuristics for the vehicle routing problem with stochastic demands. In *Parallel Problem Solving from Nature-PPSN VIII*, pages 450–460. Springer, 2004.

[8] Leonora Bianchi, Mauro Birattari, Marco Chiarandini, Max Manfrin, Monaldo Mastrolilli, Luis Paquete, Olivia Rossi-Doria, and Tommaso Schiavinotto. Hybrid metaheuristics for the vehicle routing problem with stochastic demands. *Journal of Mathematical Modelling and Algorithms*, 5(1):91–110, 2006.

[9] John R. Birge and Francois V. Louveaux. *Introduction to stochastic programming*. Springer Science & Business Media, 2011.

[10] Ann Melissa Campbell and Barrett W. Thomas. Challenges and advances in a priori routing. In *The Vehicle Routing Problem: Latest Advances and New Challenges*, pages 123–142. Springer, 2008.

[11] Abraham Charnes and William W. Cooper. Chance-constrained programming. *Management science*, 6(1):73–79, 1959.

[12] Krishna Chepuri and Tito Homem-De-Mello. Solving the vehicle routing problem with stochastic demands using the cross-entropy method. *Annals of Operations Research*, 134(1):153–181, 2005.

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

[14] Geoff Clarke and John W. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4):pp. 568–581, 1964. ISSN 0030364X.

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

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

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

[18] Merrill M. Flood. The traveling salesman problem. *Operations Research*, 4(1): 61–75, 1956.

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

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

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

[22] Michel Gendreau, Ola Jabali, and Walter Rei. Stochastic vehicle routing problems. In P. Toth and D. Vigo, editors, *Vehicle routing: Problems, methods, and applications*, MOS-SIAM Series on Optimization, chapter 8. SIAM, Philadelphia, second edition edition, 2014.

[23] Michel Gendreau, Ola Jabali, and Walter Rei. 50th anniversary invited article–future research directions in stochastic vehicle routing. *Transportation Science*, 50 (4):1163–1173, 2016.

[24] Bernard Gendron, Teodor Gabriel Crainic, Antonio Frangioni, and François Guertin. OOBB: Object-oriented tools for parallel branch-and-bound. In *PAREO*, January 16-21 2005.

[25] Chrysanthos E. Gounaris, Wolfram Wiesemann, and Christodoulos A. Floudas. The robust capacitated vehicle routing problem under demand uncertainty. *Operations Research*, 61(3):677–693, 2013.

[26] Gregory Gutin and Abraham P. Punnen. *The traveling salesman problem and its variations*, volume 12. Springer Science & Business Media, 2006.

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

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

[29] Angel A. Juan, Javier Faulin, Scott E. Grasman, Daniel Riera, J. Marull, and Carlos A. Mendez. Using safety stocks and simulation to solve the vehicle routing problem with stochastic demands. *Transportation Research Part C: Emerging Technologies*, 19(5):751–765, 2011.

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

[31] Véronique Lambert, Gilbert Laporte, and François Louveaux. Designing collection routes through bank branches. *Computers & Operations Research*, 20(7):783–791, 1993.

[32] Gilbert Laporte. Fifty years of vehicle routing. *Transportation Science*, 43(4): 408–416, 2009.

[33] Gilbert Laporte and François Louveaux. Solving stochastic routing problems with the Integer L-shaped method. In TeodorGabriel Crainic and Gilbert Laporte, editors, *Fleet Management and Logistics*, pages 159–167. Springer US, 1998. ISBN 978-0-7923-8161-7.

[34] Gilbert Laporte and François V. Louveaux. The Integer L-shaped method for stochastic integer programs with complete recourse. *Operations research letters*, 13(3):133–142, 1993.

[35] Gilbert Laporte, François V. Louveaux, and Luc Van Hamme. An Integer L-shaped algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research*, 50(3):415–423, 2002.

[36] François V. Louveaux. An introduction to stochastic transportation models. In Martine Labbé, Gilbert Laporte, Katalin Tanczos, and Philippe Toint, editors, *Operations Research and Decision Aid Methodologies in Traffic and Transportation Management*, volume 166 of *NATO ASI Series*, pages 244–263. Springer Berlin Heidelberg, 1998. ISBN 978-3-642-08428-7. doi: 10.1007/978-3-662-03514-6_11.

[37] François V. Louveaux and Juan-José Salazar-González. Exact approach for the vehicle routing problem with stochastic demands and preventive returns. Private communication, 2017.

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

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

[40] Jorge E. Mendoza, Louis-Martin Rousseau, and Juan G. Villegas. A hybrid metaheuristic for the vehicle routing problem with stochastic demand and duration constraints. *Journal of Heuristics*, pages 1–28, 2015.

[41] Clair E. Miller, Albert W. Tucker, and Richard A. Zemlin. Integer programming formulation of traveling salesman problems. *Journal of the ACM (JACM)*, 7(4): 326–329, 1960.

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

[43] Walter Rei, Michel Gendreau, and Patrick Soriano. Local branching cuts for the 0-1 integer l-shaped algorithm. Technical Report CIRRELT-2007-23, Centre Interuniversitaire de Recherche sur les Réseaux d´Entreprise, la Logistique et le Transport (CIRRELT), Montréal, Québec., 2007.

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

[45] Majid Salavati-Khoshghalb, Michel Gendreau, Ola Jabali, and Walter Rei. A hybrid recourse policy for the vehicle routing problem with stochastic demands.

Technical Report CIRRELT-2017-42, Centre Interuniversitaire de Recherche sur les Réseaux d´Entreprise, la Logistique et le Transport (CIRRELT), Montréal, Québec., July 2017.

[46] Majid Salavati-Khoshghalb, Michel Gendreau, Ola Jabali, and Walter Rei. A rule-based recourse for the vehicle routing problem with stochastic demands. Technical Report CIRRELT-2017-36, Centre Interuniversitaire de Recherche sur les Réseaux d´Entreprise, la Logistique et le Transport (CIRRELT), Montréal, Québec., July 2017.

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

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

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

[50] Réne Séguin. *Problèmes stochastiques de tournées de véhicules*. PhD thesis, Universite de Montreal, 1994.

[51] Elyn Lizeth Solano Charris. *Optimization methods for the robust vehicle routing problem*. PhD thesis, Troyes, 2015.

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

[53] Ilgaz Sungur, Fernando Ordónez, and Maged Dessouky. A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty. *IIE Transactions*, 40(5):509–523, 2008.

[54] Dušan Teodorović and Goran Pavković. A simulated annealing technique approach to the vehicle routing problem in the case of stochastic demand. *Transportation Planning and Technology*, 16(4):261–273, 1992.

[55] Paolo Toth and Daniele Vigo. *Vehicle Routing: Problems, Methods, and Applications*, volume 18. SIAM, 2014.

[56] Richard M. Van Slyke and Roger Wets. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17(4):638–663, 1969.

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

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

Appendix I

Performance of All Three Papers

The Table I.I present overall performance of all three papers in the dissertation, containing the following notations: 1) the “opt. sol.” columns present the percentage of instances that were solved to optimality in each policy; 2) the “time” columns refer to the average running times in seconds that were needed by the algorithm to solve one instance to optimality; 3) the “gap” columns present the average optimality gap obtained by the algorithm over all instances solved for each policy.

Table I.I – Performance of all three papers.

	π_1			π_2			π_3			$\{\underline{\theta},\bar{\theta}\}$={0.45, 0.55}			$\{\underline{\theta},\bar{\theta}\}$={0.35, 0.65}			$\{\underline{\theta},\bar{\theta}\}$={0.25, 0.75}			$\{\underline{\theta},\bar{\theta}\}$={0.15, 0.85}			$\{\underline{\theta},\bar{\theta}\}$={0.05, 0.95}			Optimal	
opt. sol.	time	gap	opt. sol.	time	gap	opt. sol.	time	gap	opt. sol.	time	gap	opt. sol.	time	gap	opt. sol.	time	gap	opt. sol.	time	gap	opt. sol.	time	gap	opt. sol.	time	gap
49.62%	1891.51	3.74%	51.74%	1653.38	3.04%	44.92%	2416.54	5.68%	63.86%	1332.29	0.50%	64.32%	1274.63	0.49%	64.09%	1549.79	0.53%	63.41%	1205.95	0.55%	60.23%	1616.58	0.71%	51.59%	1407.42	0.90%