An Exact Algorithm to Solve the Vehicle Routing Problem with Stochastic Demands under an Optimal Restocking Policy

Majid Salavati-Khoshghalb, Michel Gendreau, Ola Jabali, Walter Rei

PII: S0377-2217(18)30661-1
DOI: 10.1016/j.ejor.2018.07.039
Reference: EOR 15277

To appear in: *European Journal of Operational Research*

Received date: 19 September 2017
Revised date: 23 July 2018
Accepted date: 23 July 2018

Highlights

- We present an exact method to solve vehicle routing problem with stochastic demands.
- We lower approximate the optimal restocking policy to bound recourse function.
- We enhance the Integer L-shaped method by redefining lower bounding functionals.
- We can solve problems with 100 customers having arbitrary discrete demands.

An Exact Algorithm to Solve the Vehicle Routing Problem with Stochastic Demands under an Optimal Restocking Policy

Majid Salavati-Khoshghalb[a,b,*], Michel Gendreau[a,c], Ola Jabali[d], Walter Rei[a,e]

[a]*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*
[b]*Département d'informatique et de recherche operationnelle, Université de Montréal, C.P. 6128, succ. Centre-ville, Montréal, Québec H3C 3J7, Canada*
[c]*Département de mathématiques et de génie industriel, Polytechnique de Montréal, C.P. 6079, succ. Centre-ville, Montréal, Québec H3C 3J7, Canada*
[d]*Dipartimento di elettronica, informazione e bioingegneria, Politecnico di Milano, Piazza Leonardo da Vinci, 32, Milano 20133, Italy*
[e]*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, a 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 to the depot 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 the presented algorithm, several lower bounding schemes are developed to approximate the expected recourse cost.

Keywords: Routing, Stochastic demands, Optimal policy, Restocking, Partial routes, Integer L-shaped algorithm, Lower bounding functionals

*Corresponding author
Email addresses: majid.salavati@cirrelt.ca (Majid Salavati-Khoshghalb), michel.gendreau@cirrelt.net (Michel Gendreau), ola.jabali@polimi.it (Ola Jabali), walter.rei@cirrelt.ca (Walter Rei)

1. Introduction

Following the seminal paper of Dantzig & Ramser (1959), the *Vehicle Routing Problem* (VRP) has been the subject of considerable research efforts over the last decades, see Laporte (2009). The aim in VRP is to find a set of routes serving a given set of customers 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. (2014), Oyola et al. (2016), and Oyola et al. (2017). 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 (1998). 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. (1989) 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. (1990); another is the *reoptimization* approach; further details can be found in Gendreau et al. (2014). 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, Birge & Louveaux (2006) 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 (2001) and Secomandi & Margot (2009) 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 & Trudeau (1986) and implemented by Gendreau et al. (1995); Hjorring & Holt (1999); Laporte et al. (2002); Rei et al. (2010) and Jabali et al. (2014). 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 & Golden (1980) and implemented by Yang et al. (2000) and Bianchi et al. (2004).

Employing the optimal restocking policy entails simultaneously optimizing the vehicle routes and the customer-specific thresholds. As these thresholds are an outcome of the optimization, the optimal restocking policy does not directly allow a company to systematically control the risk of encountering failures. Salavati-Khoshghalb et al. (2017) and Salavati-Khoshghalb et al.

(2018) proposed different recourse policies, which allow a company to determine its customer-specific thresholds according to a number of operational rules. Salavati-Khoshghalb et al. (2017) proposed three volume based policies, which use simplistic comparisons of the vehicle's residual capacity in order to decide when PR trips are performed, e.g., executing a PR trip once the available vehicle capacity is below a preset percentage of its total capacity. Salavati-Khoshghalb et al. (2018) proposed a hybrid recourse policy which is a more advanced form of a rule-based recourse policy. The hybrid recourse policy combines risk-based and distance-based rules. For a given route, the authors define a risk measure, which computes the risk of failure at the next customer. This risk measure is then compared with two preset thresholds. Namely, the minimum restocking threshold and a maximum proceeding threshold. In the case where the risk measure is greater than the minimum restocking threshold, the vehicle executes a PR trip, whereas if the risk measure is less than the maximum proceeding threshold, the vehicle proceeds with its planned route. In the cases where the risk measure falls between the maximum proceeding threshold and the minimum restocking threshold, a distance measure employed. This measure compares the cost of performing a PR trip at the current customer with the expected failure costs, resulting from BF trips, performed at all subsequent customers in the planned route. If the cost of the former is lower than the latter a PR trip is performed, otherwise the vehicle proceeds with its planned route.

We present a small example in order to illustrate the differences between the different recourse policies (optimal restocking, classical, rule-based and hybrid). In Figure 1 the pair of numbers below each vertex specifies the coordinate of the vertex in $[0, 1000]^2$. The expected demand of each customer is shown by a red integer on the right-hand-side of the vertex. The capacity of the vehicle was set to 45. The support of the demand probability distributions of customers v_1, v_2, v_3, and v_4 are $\{11,13,15,17,19\}$, $\{6,8,10,12,14\}$, $\{11,13,15,17,19\}$ and $\{1,3,5,7,9\}$, respectively. A probability of 0.2 was associated with each of the five possible values for each customer. All four policies use the route $0 - v_4 - v_3 - v_1 - v_2 - 0$. The rule-based policy is based on the second policy proposed Salavati-Khoshghalb et al. (2017), which outperformed the other two. According to this policy, when leaving a customer on a planned route a PR trip is performed if the residual capacity of the vehicle is less than η times the expected demand of the subsequent customer on the route. This policy with $\eta = 1$ was used in the example. As for the hybrid policy the minimum restocking threshold was set to 0.65 and a maximum proceeding threshold was set to 0.35.

In Table 1 we summarize the results of the example for the four policies. For the optimal restocking policy and the rule-based policy we present the customer thresholds. We note that while the routing cost (i.e., the first stage cost) is the same for all four policies, the expected recourse costs (i.e., the second stage cost) differ from one policy to another. In particular, the classical policy incurs the highest expected recourse cost, and the rule based policy has a higher expected recourse cost than that of the optimal restocking policy. Finally, we note that in this examples the hybrid policy, albeit employing a mixed policy structure, yields the same expected recourse cost as the optimal restocking policy.

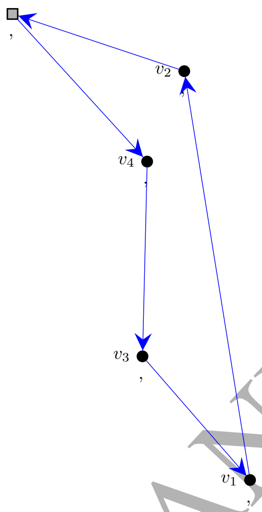


Figure 1: Small example with four customers randomly scattered in $[0, 1000]^2$.

Table 1: Comparison of four recourse policies on the instance of Figure 1

Policy	Total cost	Routing cost	Recourse cost	θ_1	θ_2	θ_3	θ_4
Classical	2881.8	2584.0	297.8				
Hybrid	2700.6	2584.0	116.6				
Optimal	2700.6	2584.0	116.6	12	0	17	15
Rule-based	2760.6	2584.0	176.6	10	0	15	15

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. (2014). 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. (1995), Hjorring & Holt (1999), Laporte et al. (2002), and Jabali et al. (2014). The column generation approach was applied to the VRPSD by Christiansen & Lysgaard (2007), as well as by Gauvin et al. (2014). All of these papers implemented the classical recourse policy. More recently, Salavati-Khoshghalb et al. (2017) and Salavati-Khoshghalb et al. (2018) 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 & Golden (1980) 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 metaheuristic solution procedures to solve the VRPSD by Yang et al. (2000) and Bianchi et al. (2004). Generally, these heuristic procedures result in overall sub-optimal pair of routing and recourse decisions.

Recently, Louveaux & Salazar-González (2017) 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 §2 lays out the VRPSD model under the a priori approach with an optimal restocking policy. We devote Section §3 to propose an exact method, for solving the VRPSD under an optimal restocking policy, enhanced by various lower bounding schemes. Section §4 presents the results of a computational study to examine the performance of the proposed exact method. Section §5 proposes some conclusions and future research directions.

2. Optimal Restocking Recourse Policy Under the A Priori Approach

In Section §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 §2.2.

2.1. VRPSD Formulation Under an A Priori Approach

This section revisits the VRPSD formulation presented by Gendreau et al. (1995) and Laporte et al. (2002). 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 $\{\xi_i^1, \xi_i^2, \dots, \xi_i^l, \dots, \xi_i^{s^i}\}$, 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. (1990). 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. (1995) and Laporte et al. (2002), the a priori model for the VRPSD is formulated as follows:

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

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

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

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

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

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

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

In this formulation, constraints (2) ensure that exactly m vehicle routes that start and end at the depot are established; constraints (3) ensure that each customer is connected to two other vertices; constraints (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 (1) as $\sum_{i<j} c_{ij} x_{ij}$.

Let us now suppose that an a priori routing solution x in model (1)-(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 (1).

Dror & Trudeau (1986) 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 (8), where $\mathcal{Q}^{r,\delta}$ denotes the expected recourse cost of the r^{th} a priori route in the orientation $\delta = 1, 2$.

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

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.

2.2. *The Optimal Restocking Policy*

In this section we recall the optimal restocking policy, devised by Yee & Golden (1980) 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. (2000); 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,

$$F_{i_j}(q) = \min \begin{cases} H_{i_j,i_{j+1}}(q) : \displaystyle\sum_{k:\xi^k_{i_{j+1}} \leq q} F_{i_{j+1}}(q - \xi^k_{i_{j+1}}) p^k_{i_{j+1}} + \\ \qquad \displaystyle\sum_{k:\xi^k_{i_{j+1}} > q} [b + 2c_{1,i_{j+1}} + F_{i_{j+1}}(Q + q - \xi^k_{i_{j+1}})] p^k_{i_{j+1}}, \\ H'_{i_j,i_{j+1}}(q) : c_{1,i_j} + c_{1,i_{j+1}} - c_{i_j,i_{j+1}} + \displaystyle\sum_{k=1}^{s_i} F_{i_{j+1}}(Q - \xi^k_{i_{j+1}}) p^k_{i_{j+1}} \end{cases} \tag{9}$$

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. (2000), since it only considers the recourse cost and not the total cost of the route. Using equation (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 & Golden (1980) and Yang et al. (2000)). 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 (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,

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

To compute the expected recourse cost of the route for the second orientation (i.e., $\mathcal{Q}^{r,2}$), we reapply function (10) to the reverse of the a priori route $\vec{v}$.

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 & Louveaux (1993) 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 & Louveaux (1993), 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 & Holt (1999) (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. (2002) for multi-VRPSD, where customer demands follow continuous distributions. Jabali et al. (2014) generalize the concept of partial routes proposed by Hjorring & Holt (1999) 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. (2014) to the case of optimal restocking policy when customers demand are defined through arbitrary discrete distributions. The LBF cuts of Jabali et al. (2014) 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 §3.1, we first revisit the Integer L-shaped algorithm. Then, in subsection §3.2 we present a lower bounding scheme to approximate $\mathcal{Q}(x)$, where x contains partial routes of Jabali et al. (2014). In subsection §3.3, we provide a general lower bound L where $L \leq \mathcal{Q}(x)$ and x satisfies (2)-(7).

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 (14). The first current problem CP^0 can be presented by (11), (2), (3),(5), (6), and (14). At an arbitrary iteration ν, CP^{ν} is shown in the following model,

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

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

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

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

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

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

where, constraints (12), (13), and (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 (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 (12) at iteration ν. We also denote by $\mathbf{ST}^{\nu-1}$ the set of index sets of the vertices violating (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 Lysgaard et al. (2004). When no violated constraint (12) is detected, the lower bounding cuts (13) are added to strength the overall bounding scheme. An exact separation procedure developed by Jabali et al. (2014) 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 (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. (2014) (see subsection §3.2 for further details). For each partial route h the functional $W_p^h(x)$ ensures that the constraint is active on relevant portions of the solution space and, is redundant otherwise (see subsection §3.2 for further details). 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 §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 (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 $\mathbf{OC}^{\nu} = \mathbf{OC}^{\nu-1} \cup \{f'\}$.

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. (2014). We note that Θ_p^q in (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 $\Theta_p^q = \sum_{h \in \mathbf{PR}^q} \Theta_p^{qh}$. We then drop the index q in Θ_p^{qh} and present it by Θ_p^h.

Generally, a partial route stems from a fractional solution and 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 $\bar{\mathcal{G}}^{\nu}$), where there is an edge (v_i, v_j) in $\bar{\mathcal{G}}^{\nu}$ if $x_{ij}^{\nu} = 1$. 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 is connected to them at unique articulation vertices. Figure 2 shows an example of three possible partial routes.

In the α-route topology the first and last chains are viewed as CVSs, while the intermediate component (containing potentially multiple chains and UVSs) is considered as a single-UVS. This topology corresponds to the one proposed by Hjorring & Holt (1999). In the β-route topology the actual alternation of CVSs and UVSs is captured. In the γ-route topology each chain is viewed as a UVS and articulation vertices are viewed as single-CVSs. The separation procedure proposed by Jabali et al. (2014) detects all chains and CVSs, which implicitly implies that a β-route topology is detected. Once this topology is detected an appropriate α-route and an appropriate γ-route may be derived.

Formally let κ denote the number of chains and $\kappa - 1$ be the number of UVSs in partial route. Let $S_h^t = \{v_{h_1}^t, \ldots, v_{h_l}^t\}$ be the t^{th} chain in partial route $h \in \mathbf{PR}^{\nu}$, where $v_{h_z}^t$ is the z^{th} vertex in S_h^t, and h_l is the number of vertices in S_h^t. Therefore,

$$\sum_{(v_i, v_j) \in S_h^t} x_{ij}^{\nu} = |S_h^t| - 1, \ \forall t = 1, \ldots, \kappa. \tag{16}$$

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

$$\sum_{v_i, v_j \in U_h^t} x_{ij}^{\nu} = |U_h^t| - 1, \ \forall t = 1, \ldots, \kappa - 1. \tag{17}$$

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

$$\sum_{v_j \in U_h^t} x_{h_l^t, j}^{\nu} = 1, \ \forall t \le \kappa - 1, \tag{18}$$

and

$$\sum_{v_j \in U_h^{t-1}} x^{\nu}_{h_1^t, j} = 1, \ \forall t \geq 2 \tag{19}$$

For completeness we recall the definition of the functional $W_p^h(x)$, as introduced by Jabali et al. (2014):

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

We now describe an approximation technique to compute Θ_p^h in order to add LBF cuts (13). In (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. 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.

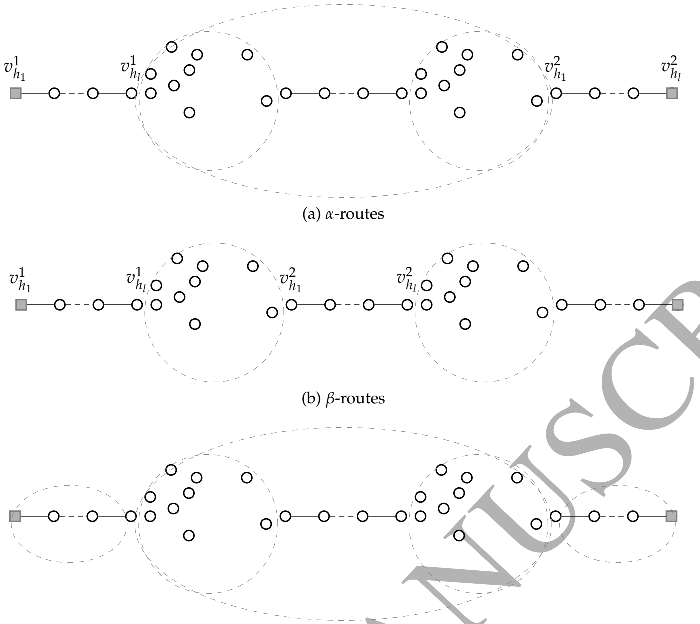
For the sake of simplicity, we redefine the partial route h, in similar terms as a route, as follows

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

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,

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

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



(a) α-routes

(b) β-routes

(c) γ-routes

Figure 2: Three partial route topologies (adapted from Salavati-Khoshghalb et al. (2017))

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 ${i_a}^{\text{th}}$ 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 (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 $\mathrm{s} = (\boxed{}_{i_a}, q)$ represent the state of the system after serving the (still unknown) customer in the ${i_a}^{\text{th}}$ 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 ${i_{a+1}}^{\text{th}}$ position (i.e., $\square_{i_{a+1}} := v_{u_1}$ in s'). We can then write

$$\hat{F}_{i_a}(\mathsf{s} = (\square_{i_a}, q), \mathsf{s}' = (v_{u_1}, q')) = \\ = \min \begin{cases} \sum_{k:\xi_{u_1}^k \leq q} \widetilde{F}_{i_{a+1}}(\mathsf{s}' = (v_{u_1}, q' := q - \xi_{u_1}^k))p_{u_1}^k + \\ \sum_{k:\xi_{u_1}^k > q} [b + 2c_{1,u_1} + \widetilde{F}_{i_{a+1}}(\mathsf{s}' = (v_{u_1}, q' := Q + q - \xi_{u_1}^k))]p_{u_1}^k, \\ c_{1,i_a} + c_{1,u_1} - c_{i_k,u_1} + \sum_{k=1}^{s_{u_1}} \widetilde{F}_{i_{a+1}}(\mathsf{s}' = (v_{u_1}, q' := Q - \xi_{u_1}^k))p_{u_1}^k. \end{cases} \tag{22}$$

To compute $\hat{F}_{i_a}(\mathsf{s} = (\square_{i_k}, q), \mathsf{s}' = (v_{u_1}, q'))$ in (22), the PR trip travel cost is replaced by a lower bound $\underset{v_{u_e} \in U_h^1 : v_{u_e} \neq v_{u_1}}{minimum} \{c_{1,u_e} + c_{1,u_1} - c_{u_e,u_1}\}$. 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

$$\widetilde{F}_{i_a}(\mathsf{s} = (\square_{i_a}, q)) = \underset{v_{u_e} \in U_h^1}{min} \hat{F}_{i_a}(\mathsf{s} = (\square_{i_a}, q), \mathsf{s}' = (v_{u_e}, q')). \tag{23}$$

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 $\underset{v_{u_e} \in U_h^1}{minimum}\{c_{1,u_e} + c_{1,i_{j+1}} - c_{u_e,i_{j+1}}\}$. 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 (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 (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

$$\Theta^h_\alpha = \min\{\widetilde{F}^1_{i_1}(Q), \widetilde{F}^2_{i_1}(Q)\} \tag{24}$$

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 (13).

3.3. General Lower Bound

In this subsection, we propose a procedure to obtain a general lower bound L to be used in constraints (13) and (14). As defined by Laporte & Louveaux (1993), the expected recourse cost associated to the feasible solution x^L with minimum expected recourse cost corresponds to a general lower bound. Laporte & Louveaux (1998) 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 & Louveaux (1998) is further improved by Laporte et al. (2002). 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 & Louveaux (1993),

$$L = \mathcal{Q}(x^L) \le \min_x\{\mathcal{Q}(x)|(2)-(6)\} = \sum_{k=1}^{m} \min\{\mathcal{Q}^{k,1}(\vec{v}^k), \mathcal{Q}^{k,2}(\vec{v}^k)\}. \tag{25}$$

For computing L in (25), we assume that: the vehicle route denoted by $\vec{v}^{12}$ is obtained by concatenating $\vec{v}^2$ after $\vec{v}^1$; v_{l^1} and v_{f^2} 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; $\bar{F}^{\vec{v}^{12}}_{v_{l^1}}(.)$ and $F^{\vec{v}^{12}}_{v_{l^1}}(.)$ are the expected recourse costs from the depot to v_{l^1} and expected cost-to-go from v_{l^1} to the depot going through $\vec{v}^2$, respectively; and $p^q_{v_{l^1}}$ is the probability of having q units of residual capacity after serving customer v_{l^1}.

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

$$F^{\vec{v}^{12}}_{v_1}(Q) = \sum_q \{\bar{F}^{\vec{v}^{12}}_{v_{l^1}}(q) + F^{\vec{v}^{12}}_{v_{l^1}}(q)\} p^q_{v_{l^1}}. \tag{26}$$

By definition, we have

$$F_{v_{l1}}^{\vec{v}^{12}}(q) = \min \begin{cases} \sum_{k:\xi_{v_{f2}}^k \le q} F_{v_{f2}}^{\vec{v}^{12}}(q - \xi_{v_{f2}}^k) p_{v_{f2}}^k + \\ \sum_{k:\xi_{v_{f2}}^k > q} [b + 2c_{1,v_{f2}} + F_{v_{f2}}^{\vec{v}^{12}}(Q + q - \xi_{v_{f2}}^k)] p_{v_{f2}}^k, \\ c_{1,v_{l1}} + c_{1,v_{f2}} - c_{v_{l1},v_{f2}} + \sum_{k=1}^{s_{v_{f2}}} F_{v_{f2}}^{\vec{v}^{12}}(Q - \xi_{v_{f2}}^k) p_{v_{f2}}^k. \end{cases} \tag{27}$$

We also have $F_{v_{l1}}^{\vec{v}^{12}}(q) \le c_{1,v_{l1}} + c_{1,v_{f2}} - c_{v_{l1},v_{f2}} + \sum_{k=1}^{s_{v_{f2}}} F_{v_{f2}}^{\vec{v}^{12}}(Q - \xi_{v_{f2}}^k) p_{v_{f2}}^k$ which coupled with (26) results in

$$F_{v_1}^{\vec{v}^{12}}(Q) \le \sum_q \{ \bar{F}_{v_{l1}}^{\vec{v}^{12}}(q) + c_{1,v_{l1}} + c_{1,v_{f2}} - c_{v_{l1},v_{f2}} + \sum_{k=1}^{s_{v_{f2}}} F_{v_{f2}}^{\vec{v}^{12}}(Q - \xi_{v_{f2}}^k) p_{v_{f2}}^k \} p_{v_{l1}}^q. \tag{28}$$

Assuming that $\vec{v}^{12}$ is equivalent to the concatenation of $\vec{v}^1$ and $\vec{v}^2$, the relation (28) can further yield

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

where, the first term in (28) is equivalent to $F_{v_1}^{\vec{v}^1}(Q)$ in the backward fashion and the last term in (28) is equivalent to $F_{v_1}^{\vec{v}^2}(Q)$ 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:

$$F_{v_1}^{\vec{v}^{1\ldots m}}(Q) \le \sum_{k=1}^{m-1} c_{\mathrm{PR}}^k + \sum_{k=1}^{m} F_{v_1}^{\vec{v}^k}(Q) \tag{29}$$

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 $\vec{v}^1, \vec{v}^2, \ldots, \vec{v}^m$, 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 (29) consists of all customers. To calculate a general lower bound $L^* \le L$, we can approximate the left-hand-side of (29) 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.,

$$\tilde{l}_z = (v_1 = v_{i_1}, \square_{i_2}, \square_{i_3}, \ldots, \square_{i_{t-1}}, v_z, v_{i_{t+1}} = v_1). \tag{30}$$

This is done exactly as in §3.2. Finally, a general lower bound L^* can be computed as

$$L^* = \min_{z:2,\ldots,n} \widetilde{F}_{v_1}^{\bar{l}_z}(Q) - \sum_{k=1}^{m-1} c_{\mathrm{PR}}^k. \tag{31}$$

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. (2017)), 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 2. 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 2: 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 $b = \sum_{i=2,\dots,n} c_{i1}/(n-1)$ is incurred when experiencing route failures. We recall that b primarily penalizes disruption at a customer location caused by the second vehicle visit.

The Instances Generated by Louveaux & Salazar-González (2017): The instances of Louveaux & Salazar-González (2017) 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 $k = 1, \dots, K$, 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 vehicles 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 & Salazar-González (2017) considered a fixed cost of $\Delta = 0, 10, 100$ for the 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 dissatisfaction in the failure events.

The Integer L-shaped algorithm and the bounding scheme are coded in C++ using ILOG CPLEX 12.6. The subtour elimination and capacity constraints (4) are identified using the CVRPSEP package of Lysgaard et al. (2004). The general branch-and-cut framework as the Integer L-shaped algorithm is implemented using the OOBB package developed by Gendron et al. (2005). 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.

In subsection 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 by pricing the optimal solutions under the classical policy. In subsection 4.2, we report the results obtained by the proposed algorithm on the specialized instances generated by Louveaux & Salazar-González (2017), in which all customer demands follow identical distributions.

4.1. Quality of the Integer L-Shaped Algorithm

We now present the computational result, expressing the performance of the proposed exact algorithm in Tables 3 and 5 for symmetric and asymmetric instances. The conducted experiments are aggregated according to the pair (n, m) and the filling coefficient $\bar{f}$. Tables 3 and 5 report the following information:

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) the "$\leq$ 1%" 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 3 and 5, we observe similar trends that were reported by Gendreau et al. (1995), Laporte et al. (2002), and Jabali et al. (2014) 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 shows overall 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 3 and 5, the Integer L-shaped algorithm implemented in this paper optimally solves 227 out 440 symmetric instances and 242 out of the 440 asymmetric instances; which 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 instances with an optimality gap $\leq$ 1% of the symmetric 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 policy. Tables 4 and 6 illustrate the comparisons of two recourse policies with respect to the total cost denoted by "Sav1"$= \frac{Q^{class.}(x^*_{opt}) - Q^{opt}(x^*_{opt})}{cx^*_{opt} + Q^{class.}(x^*_{opt})} \times 100$ and the expected recourse cost as "Sav2"$= \frac{Q^{class.}(x^*_{opt}) - Q^{opt}(x^*_{opt})}{Q^{class.}(x^*_{opt})} \times 100$, in which x^*_{opt} is obtained by optimally solving a VRPSD instance under optimal restocking policy. The solution x^*_{opt} has a first stage cost of cx^*_{opt} and an expected recourse cost of $Q^{opt}(x^*_{opt})$. Furthermore, $Q^{class.}(x^*_{opt})$ is the expected recourse cost of optimal routing decision x^*_{opt}. It should be noted that the classical recourse policy consists of following the planned route and performing BF and restocking trips at failures and exact stockouts, respectively. The weighted average savings in terms of "Sav1" are 0.65% and 0.61% for the symmetric and asymmetric 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 it with two other policies from the literature. The first is the rule-based policy proposed by Salavati-Khoshghalb et al. (2017), which entails that a PR trip is performed if the residual capacity of the vehicle is less than $\eta\bar{\xi}$, where $\bar{\xi}$ is the expected demand of the subsequent

Table 3: Performance measures for the 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 4: Average savings vs classical recourse with respect to total cost (Sav1), and recourse cost (Sav2) for the symmetric instances

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: Performance measures for the 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 6: Average savings vs classical recourse with respect to total cost (Sav1), and recourse cost (Sav2) for the asymmetric instances

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%

Table 7: Average savings vs rule-based recourse policy with $\eta\bar{\xi}$ for $\eta = 1$

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 8: Average savings vs hybrid recourse policy for $\underline{\theta}$-$\bar{\theta}$: $0.35 - 0.65$

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%

customer on the route. Salavati-Khoshghalb et al. (2017) achieved the best results by setting η to one. We therefore compare the optimal policy with these results. The second policy is the hybrid policy proposed by Salavati-Khoshghalb et al. (2018), where the best results were obtained by setting the maximum proceeding threshold, denoted by $\underline{\theta}$, to 0.35, and the minimum restocking threshold, denoted by $\bar{\theta}$ to 0.65. We therefore compare the optimal policy with these results. Tables 7 and 8 express the 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 the optimal routing decisions obtained by solving the VRPSD instances under the optimal restocking policy, the best rule-based and the hybrid recourse policies, respectively. As presented in Tables 7 and 8, the best rule-based policy displays less deviation from the optimal restocking policy. The latter observation provides insights in the structure of the optimal restocking policy, which further imply that this policy can be approximated more efficiently in terms of the quality (here the total costs) of the optimal routing solution by rule-based policies designed by Salavati-Khoshghalb et al. (2017).

In order to compare the solution structures between the various policies, we used the Hamming distance. We recall that the Hamming distance with respect to a reference solution $\bar{x}$ is computed as follows:

$$\Delta(x, \bar{x}) = \sum_{(v_i,v_j)\in\bar{T}} (1 - x_{ij}) + \sum_{(v_i,v_j)\in E\setminus\bar{T}} x_{ij} \tag{32}$$

Table 9: Average Hamming distance between the optimal recourse solutions and the rule-based recourse policy with $\eta\bar{\xi}$ for $\eta = 1$

n	m	$\bar{f}$	solved	Hamm. Dist.	$\bar{f}$	solved	Hamm. Dist.	$\bar{f}$	solved	Hamm. Dist.	$\bar{f}$	solved	Hamm. Dist.
20	2	0.90	10	42	0.92	10	42	0.94	10	41	0.96	10	41
30	2	0.90	10	29	0.92	8	30	0.94	10	29	0.96	7	25
40	2	0.90	10	22	0.92	10	22	0.94	10	22	0.96	6	21
40	3	0.90	5	50	0.92	7	49	0.94	4	47	0.96	2	49
40	4	0.90	0	0	0.92	0	0	0.94	0	0	0.96	0	0
50	2	0.90	10	18	0.92	8	18	0.94	10	16	0.96	4	17
50	3	0.90	4	41	0.92	4	36	0.94	3	41	0.96	1	42
50	4	0.90	2	70	0.92	1	70	0.94	0	0	0.96	0	0
60	2	0.90	10	12	0.92	9	15	0.94	7	15	0.96	6	14
60	3	0.90	3	30	0.92	1	30	0.94	1	31	0.96	0	0
60	4	0.90	0	0	0.92	0	0	0.94	0	0	0.96	0	0

Table 10: Average Hamming distance between the optimal recourse solutions and the hybrid policy with $\underline{\theta}$-$\bar{\theta}$: 0.35 – 0.65)

n	m	$\bar{f}$	solved	Hamm. Dist.	$\bar{f}$	solved	Hamm. Dist.	$\bar{f}$	solved	Hamm. Dist.	$\bar{f}$	solved	Hamm. Dist.
20	2	0.90	10	41	0.92	10	39	0.94	10	40	0.96	10	39
30	2	0.90	10	29	0.92	8	29	0.94	10	26	0.96	7	26
40	2	0.90	10	22	0.92	10	20	0.94	10	18	0.96	6	17
40	3	0.90	5	50	0.92	7	49	0.94	4	47	0.96	2	29
40	4	0.90	0	0	0.92	0	0	0.94	0	0	0.96	0	0
50	2	0.90	10	18	0.92	8	19	0.94	10	15	0.96	4	16
50	3	0.90	4	41	0.92	4	38	0.94	3	34	0.96	1	40
50	4	0.90	2	66	0.92	1	70	0.94	0	0	0.96	0	0
60	2	0.90	10	12	0.92	9	16	0.94	7	16	0.96	6	12
60	3	0.90	3	29	0.92	1	31	0.94	1	31	0.96	0	0
60	4	0.90	0	0	0.92	0	0	0.94	0	0	0.96	0	0

where, $\bar{T} = \{(v_i, v_j) \in \mathcal{E} | \bar{x}_{ij} = 1\}$.

In Table 9 we report the average Hamming distance where $\bar{x} = x^*_{opt}$ and $x = x^*_{rule}$. In Table 10 we report the average Hamming distance where $\bar{x} = x^*_{opt}$ and $x = x^*_{hybrid}$. In both these tables we only consider instances that were solved to optimality by all three policies. Furthermore, since the stochastic solution is effectively a directed solution, all computations in tables 9 and 10 are based on the directed solutions. As observed in Tables 7 and 8 the cost differences between solutions of the three policy were relatively low. However, Tables 9 and 10 show that indeed on average the solution structures of the rule based policy and the hybrid policy may be substantially different from those of the optimal policy.

4.2. *The instances Generated by Louveaux & Salazar-González (2017)*

We have compared the solutions that we obtain with those of Louveaux & Salazar-González (2017) 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.

Furthermore, it is observed from Tables 11-13 that the 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 in Louveaux and Salazar-González's implementation.

5. Conclusions

In this paper, we developed an exact solution methodology to solve the VRPSD under an optimal restocking policy. To do so, the Integer *L*-shaped algorithm was adapted. To enhance the efficiency of the Integer *L*-shaped algorithm, various lower bounding schemes were developed. The key element for successfully employing such bounding procedures is to provide effective lower approximation of the expected recourse cost of partial routes. In addition, a general lower bound enhancing the Integer *L*-shaped algorithm was also developed.

Using the exact method proposed in this paper, we were 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 is the first to solve the VRPSD under an optimal restocking policy when considering instances where customer demands follow arbitrary discrete distributions. The numerical results presented in this paper show that the resulting routes from the optimal restocking policy yield a appreciable amount of savings when compared to executing the classical policy on the same routes.

Further research in this area could focus on the exploration of the potential of applying column generation and branch and price to the considered problem. It would also be interesting to investigate how more collaborative recourse policies (where several vehicles coordinate to react to high demand situations) could be applied to the VRPSD.

Acknowledgments

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

References

References

Bertsimas, D. J., Jaillet, P., & Odoni, A. R. (1990). A priori optimization. *Operations Research*, *38*, 1019–1033.

Bianchi, L., Birattari, M., Chiarandini, M., Manfrin, M., Mastrolilli, M., Paquete, L., Rossi-Doria, O., & Schiavinotto, T. (2004). Metaheuristics for the vehicle routing problem with stochastic demands. In X. Yao, E. K. Burke, J. A. Lozano, J. Smith, J. J. Merelo-Guervós, J. A. Bullinaria, J. E. Rowe, P. Tiňo, A. Kabán, & H.-P. Schwefel (Eds.), *Parallel Problem Solving from Nature - PPSN VIII* (pp. 450–460). Berlin, Heidelberg: Springer Berlin Heidelberg.

Table 11: Louveaux & Salazar-González (2017) with $\Delta = 0$.

Instance	Instance Veh.	$\bar{f}$	Scen.	Our result Node	Run(min)	Gap	Routing	Recourse	OptRestock	Classical	L	Louveaux & Salazar-González (2017) 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 12: Louveaux & Salazar-González (2017) with $\Delta = 10$.

Instance	Instance Veh.	$\bar{f}$	Scen.	Our result Node	Run(min)	Gap	Routing	Recourse	OptRestock	Classical	L	Louveaux & Salazar-González (2017) 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 13: Louveaux & Salazar-González (2017) with $\Delta = 100$.

Instance	Instance Veh.	$\bar{f}$	Scen.	Our result Node	Run(min)	Gap	Routing	Recourse	OptRestock	Classical	L	Louveaux & Salazar-González (2017) 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

Birge, J., & Louveaux, F. V. (2006). *Introduction to Stochastic Programming*. Springer Science & Business Media.

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

Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, *6*, 80–91.

Dror, M., Laporte, G., & Trudeau, P. (1989). Vehicle routing with stochastic demands: Properties and solution frameworks. *Transportation Science*, *23*, 166–176.

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

Gauvin, C., Desaulniers, G., & Gendreau, M. (2014). A branch-cut-and-price algorithm for the vehicle routing problem with stochastic demands. *Computers & Operations Research*, *50*, 141 – 153.

Gendreau, M., Jabali, O., & Rei, W. (2014). Stochastic vehicle routing problems. In P. Toth, & D. Vigo (Eds.), *Vehicle routing: Problems, methods, and applications* MOS-SIAM Series on Optimization chapter 8. (pp. 213–239). Philadelphia: SIAM. (second edition ed.).

Gendreau, M., Laporte, G., & Sguin, R. (1995). An exact algorithm for the vehicle routing problem with stochastic demands and customers. *Transportation Science*, *29*, 143–155.

Gendron, B., Crainic, T. G., Frangioni, A., & Guertin, F. (2005). Oobb: Object-oriented tools for parallel branch-and-bound. In *PAREO*.

Hjorring, C., & Holt, J. (1999). New optimality cuts for a single-vehicle stochastic routing problem. *Annals of Operations Research*, *86*, 569–584.

Jabali, O., Rei, W., Gendreau, M., & Laporte, G. (2014). Partial-route inequalities for the multi-vehicle routing problem with stochastic demands. *Discrete Applied Mathematics*, *177*, 121 – 136.

Laporte, G. (2009). Fifty years of vehicle routing. *Transportation Science*, *43*, 408–416.

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

Laporte, G., & Louveaux, F. V. (1998). Solving stochastic routing problems with the Integer L-shaped method. In T. G. Crainic, & G. Laporte (Eds.), *Fleet Management and Logistics* (pp. 159–167). Boston, MA: Springer US.

Laporte, G., Louveaux, F. V., & Van Hamme, L. (2002). An Integer L-shaped algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research*, *50*, 415–423.

Louveaux, F. (1998). An introduction to stochastic transportation models. In M. Labbé, G. Laporte, K. Tanczos, & P. Toint (Eds.), *Operations Research and Decision Aid Methodologies in Traffic and Transportation Management* (pp. 244–263). Berlin, Heidelberg: Springer Berlin Heidelberg.

Louveaux, F. V., & Salazar-González, J.-J. (2017). Exact approach for the vehicle routing problem with stochastic demands and preventive returns. Private communication.

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

Oyola, J., Arntzen, H., & Woodruff, D. L. (2016). The stochastic vehicle routing problem, a literature review, part I: models. *EURO Journal on Transportation and Logistics*. doi:10.1007/s13676-016-0100-5.

Oyola, J., Arntzen, H., & Woodruff, D. L. (2017). The stochastic vehicle routing problem, a literature review, part II: solution methods. *EURO Journal on Transportation and Logistics*, *6*, 349–388.

Rei, W., Gendreau, M., & Soriano, P. (2010). A hybrid monte carlo local branching algorithm for the single vehicle routing problem with stochastic demands. *Transportation Science*, *44*, 136–146.

Salavati-Khoshghalb, M., Gendreau, M., Jabali, O., & Rei, W. (2017). *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.

Salavati-Khoshghalb, M., Gendreau, M., Jabali, O., & Rei, W. (2018). A hybrid recourse policy for the vehicle routing problem with stochastic demands. *EURO Journal on Transportation and Logistics*. doi:10.1007/s13676-018-0126-y.

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

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

Yang, W.-H., Mathur, K., & Ballou, R. H. (2000). Stochastic vehicle routing problem with restocking. *Transportation Science*, *34*, 99–112.

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