CIRRELT

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

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

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

Majid Salavati-Khoshghalb
Michel Gendreau
Ola Jabali
Walter Rei

July 2017

CIRRELT-2017-42

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

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

www.cirrelt.ca

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

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

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

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

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

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

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

Abstract. In this paper we 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% 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, integer L-shaped algorithm, lower bounding functionals.

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

---

* Corresponding author: majid.salavati @cirrelt.ca

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

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 (2014) 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 (1998)). Therefore, an ever growing class of problems, referred to as stochastic vehicle routing problem (SVRP), has been receiving increasing attention (Gendreau et al. (2016)). 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 (2005)), garbage collection (Yang et al. (2000)) and the collection of money from banks (Lambert et al. (1993)).

Several modelling paradigms have been proposed for the VRPSD, see Gendreau et al. (2014) for an extensive review. In this paper we use the *a priori* modelling paradigm, which was originally put forward by Bertsimas et al. (1990). 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 (1986)) 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. (1995), Hjorring and Holt (1999). 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 (1980) and Yang et al. (2000). 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 (Kovacs et al. (2014)).

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. (2014)). Several exact algorithms have been proposed for the VRPSD with the classical recourse. Gendreau et al. (1995), Laporte et al. (2002), and Jabali et al. (2014) use the *L*-shaped algorithm while, Christiansen and Lysgaard (2007) and Gauvin et al. (2014) use column generation approaches. Heuristic algorithms were also proposed for this problem, e.g., Gendreau et al. (1996), Rei et al. (2010), and Mendoza et al. (2015).

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 (1980). Several heuristic algorithms are proposed for this policy. A cyclic heuristic (Bertsimas et al. (1995)), a local search heuristic (Yang et al. (2000)), and a metaheuristic (Bianchi et al. (2004)).

The third recourse policy is the *rule-based recourse* policy, which was recently coined by Salavati-Khoshghalb et al. (2017). 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 (Secomandi and Margot (2009)) 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 (2009) 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. (2017) 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. (2017) 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 §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 §3. Numerical experiments are presented in Section §4. Finally, we present our conclusions and future research directions in Section §5.

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

In section §2.1, we present the two-stage stochastic programming formulation for the VRPSD, initially proposed by Laporte et al. (2002). We then present a concise taxonomy for the rule-based policies in Section §2.2. Based on this taxonomy we elaborate the proposed hybrid recourse policy in Section §2.3.

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. (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 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 $\{\xi_i^1, \xi_i^2, \ldots, \xi_i^{s^i}\}$, 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. (2002) and Jabali et al. (2014), 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:

$$\begin{aligned}
\underset{x}{\text{minimize}} \quad & \sum_{i<j} c_{ij}x_{ij} + \mathcal{Q}(x) && && (1)\\
\text{subject to} \quad & \sum_{j=2}^{n} x_{1j} = 2m, && && (2)\\
& \sum_{i<k} x_{ik} + \sum_{k<j} x_{kj} = 2, && k = 2, \dots, n && (3)\\
& \sum_{v_i,v_j \in S} x_{ij} \leq |S| - \left\lceil \frac{\sum_{v_i \in S} \mathbb{E}(\xi_i)}{Q} \right\rceil, && (S \subset \mathcal{V} \setminus \{v_1\}; 2 \leq |S| \leq n-2) && (4)\\
& 0 \leq x_{ij} \leq 1, && 2 \leq i < j < n && (5)\\
& 0 \leq x_{1j} \leq 2, && j = 2, \dots, n && (6)\\
& x = (x_{ij}), && \text{integer} && (7)
\end{aligned}$$

The objective function (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 (2) and (3) establish the degree of the vertices. Constraints (4) eliminate subtours, and ensure that the total expected demand of each route is less or equal to Q. Finally, constraints (5), (6) and (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,

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

The computation of $\mathcal{Q}^{r,\delta}$ is elaborated in section §2.3.

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.

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 $r = (v_1 = v_{r_1}, v_{r_2}, \ldots, v_{r_{l-1}}, v_{r_l} = v_1)$, 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. (2000), 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,

$$\mathbb{P}[\xi_{r_{j+1}} > \tilde{q}] = \sum_{l:\xi^l_{r_{j+1}} > \tilde{q}} p^l_{r_{j+1}} \tag{9}$$

where, the right-hand-side of equation (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 (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 with 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,

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

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 $\mathrm{DP}_{r_j}(\tilde{q})$ as follows,

$$\mathrm{DP}_{r_j}(\tilde{q}) := \begin{cases} \texttt{True} & \text{if } c_{1r_j} + c_{1r_j+1} < c_{r_jr_j+1} + (2\bar{c}_{r_j} + b)p^*_{r_j}(\tilde{q}) \\ \texttt{False} & \text{otherwise} \end{cases} \tag{10}$$

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,

$$\begin{aligned} Q^R_{r_j} = &\left\{\tilde{q} \in \{0, 1, \ldots, Q\} | \mathbb{P}[\xi_{r_j+1} > \tilde{q}] \geq \bar{\theta}\right\} \bigcup \\ &\left\{\tilde{q} \in \{0, 1, \ldots, Q\} | \underline{\theta} < \mathbb{P}[\xi_{r_j+1} > \tilde{q}] < \bar{\theta} \wedge \mathrm{DP}_{r_j}(\tilde{q})\right\} \end{aligned} \tag{11}$$

and

$$\begin{aligned} Q^P_{r_j} = &\left\{\tilde{q} \in \{1, \ldots, Q\} | \mathbb{P}[\xi_{r_j+1} > \tilde{q}] \leq \underline{\theta}\right\} \bigcup \\ &\left\{\tilde{q} \in \{1, \ldots, Q\} | \underline{\theta} < \mathbb{P}[\xi_{r_j+1} > \tilde{q}] < \bar{\theta} \wedge \overline{\mathrm{DP}}_{r_j}(\tilde{q})\right\}. \end{aligned} \tag{12}$$

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,

$$F_{r_j}(q) = \mathbb{E}_{\xi_{r_j}}\left[F^{post}_{r_j}(\tilde{q})\right] \; \forall \tilde{q} = q - \xi_{r_j}, \tag{13}$$

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.

$$F^{post}_{r_j}(\tilde{q}) = \begin{cases} b + 2c_{1r_j} + F_{r_{j+1}}(Q + \tilde{q}) & \text{if } \tilde{q} < 0 \quad (14a) \\ c_{1r_j} + c_{1r_j+1} - c_{r_jr_j+1} + F_{r_{j+1}}(Q) & \text{if } \tilde{q} \in Q^R_{r_j} \quad (14b) \\ F_{r_{j+1}}(\tilde{q}) & \text{if } \tilde{q} \in Q^P_{r_j} \quad (14c) \end{cases}$$

Using the equations (13), (14a), (14b), and (14c), the expected recourse cost in the first direction (i.e., $\delta = 1$) is as follows,

$$\mathcal{Q}^{r,1} = F_{r_1}(Q). \tag{15}$$

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 (15).

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 (1993) 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 (1969) for continues stochastic programs, which itself was based on the application of Benders decomposition to stochastic programming, see Benders (1962). In Section §3.1 we briefly present the integer *L*-shaped algorithm. Similar to Jabali et al. (2014), we use a series of lower bounding functionals (LBFs) based on general partial routes. In section §3.2 we present the concept of general partial routes and we present the structure of the LBFs. We note that section §3.2 is largely based on Jabali et al. (2014), and is presented in this paper for the sake of completeness. In section §3.3 we develop bounds specific to our hybrid recourse policy, which are used in the LBFs.

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) and (7) are relaxed, the recourse function $Q(x)$ is replaced by variable Θ, and a general lower bounding constraint (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,

$$CP^0 : \min_{x,\Theta} \quad \sum_{i<j} c_{ij} x_{ij} + \Theta \tag{1}$$

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

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

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

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

$$L \leq \Theta. \tag{16}$$

The algorithm proceeds by adding three types of constraints until optimality is guaranteed: (i) violated constraints (4) are gradually added when detected; (ii) valid inequalities

$$L + (\Theta_p - L) W(x) \leq \Theta, \qquad \forall p = \{\alpha, \beta\},\ x \text{ is a partial solution} \tag{17}$$

which are elaborated in Section §3.3, are added when encountered; and (iii) optimality cuts

$$\sum_{\substack{1 \leq i \leq j \\ x_{ij}^{\nu} = 1}} x_{ij} \leq \sum_{1 \leq i \leq j} x_{ij}^{\nu} - 1, \tag{18}$$

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 (5).

The integer L-shaped algorithm was first used by Gendreau et al. (1995) 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 (1999) 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. (2002). Jabali et al. (2014) generalized the structure of partial routes to generate several families of LBFs. It is worth noting that since Laporte et al. (2002) and Jabali et al. (2014) used LBFs for the VRPSD with classical recourse, the bound Θ_p was computed in all cases as defined in Hjorring and Holt (1999). In this paper, we use the LBFs of Jabali et al. (2014) for the VRPSD and develop a specific bound Θ_p that is applicable for the proposed hybrid policy.

3.2 General Partial Routes

LBFs (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. (2014). An illustration of a general partial route can be found in Figure (1), where the depot is duplicated for presentation convenience. We define $\bar{\mathcal{G}}^{\nu}$ 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. (2014). 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 $\bar{\mathcal{G}}^{\nu}$.
2. *Unstructured components* whose vertex set are called unstructured vertex sets (UVSs).

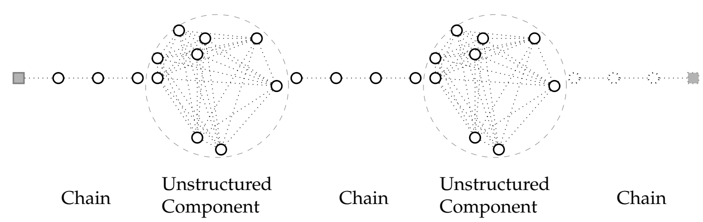
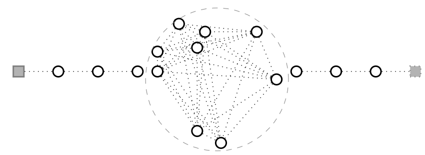


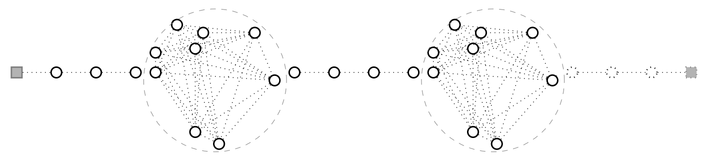
Figure 1: A general partial route h composed of sequenced and unsequenced sets.

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 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 (2). These types are emerging from the original partial route shown in Figure (1), they are denoted by α and β and they are depicted in Figures (2a) and (2b), respectively. An α-route corresponds to the initial partial route proposed by Hjorring and Holt (1999). The β-route was proposed by Jabali et al. (2014). This type of partial route maintains the exact alternation of CVSs and UVSs.



(a) α-routes



(b) β-routes

Figure 2: Generalized partial routes redefined by different views from Figure (1).

The functional $W_h(x)$ in LBFs (17) was introduced by Jabali et al. (2014) for generalized partial routes shown in Figure (2), and is defined as follows,

$$\begin{aligned} W_h(x) = {} & \sum_{t=1}^{b} \sum_{\substack{(v_i,v_j)\in S_h^t \\ v_i\neq v_1}} 3x_{ij} + \sum_{(v_1,v_j)\in S_h^1} x_{1j} + \sum_{(v_1,v_j)\in S_h^b} x_{1j} + \sum_{t=1}^{b-1} \sum_{v_i,v_j\in U_h^t} 3x_{ij} \\ & + \sum_{t=1}^{b-1} \sum_{\substack{v_j\in U_h^t \\ v_{h_l}^t\neq v_1}} 3x_{h_l^t j} + \sum_{t=2}^{b} \sum_{\substack{v_j\in U_h^{t-1} \\ v_{h_1}^t\neq v_1}} 3x_{h_1^t j} + \sum_{\substack{v_j\in U_h^1 \\ v_{h_l}^1 = v_1}} x_{h_l^1 j} + \sum_{\substack{v_j\in U_h^{b-1} \\ v_{h_1}^b = v_1 \\ v_{h_1}^{b-1}\neq v_1}} x_{h_1^b j} \\ & - (3|R_h| - 5) \end{aligned} \tag{19}$$

The proof of validity of equation (17) can be found in Jabali et al. (2014). In the coming section we develop the bound Θ_p for the VRPSD with the hybrid policy.

3.3 Bounding the Recourse Cost

We now describe the computation of Θ_p^h, 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_α^h. This derivation can then be generalized to the computation of Θ_β^h, 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

$$h = (v_1 = v_{h_1}^1, \ldots, v_{h_{|S_h^1|}}^1, U_h^1, v_{h_1}^2, \ldots, v_{h_{|S_h^2|}}^2 = v_1),$$

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

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

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,

$$\tilde{h} = (v_1 = v_{r_1}, \ldots, v_{r_{j-l}}, \square_{r_{j-l+1}}, \square_{r_{j-l+2}}, \ldots, \square_{r_j}, v_{r_{j+1}}, \ldots, v_{r_{t+1}} = v_1), \tag{20}$$

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,

$$F_{r_k}(q) = \mathbb{E}_{\xi_{r_k}}\left[F_{r_k}^{post}(q - \xi_{r_j})\right] = \mathbb{E}_{\xi_{r_k}}\left[F_{r_k}^{post}(\tilde{q})\right],\ \forall \tilde{q} = q - \xi_{r_k}. \tag{13}$$

For the sake of simplicity, we use the notation $F_{r_k}(.)$ as defined in (14) 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 (14). 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 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 3.2 we then bound the onward recourse cost from the $j - l + 1^{\text{th}}$ customer. We recall that

$$F_{r_j}(q) = \mathbb{E}_{\xi_{r_j}}\left[F_{r_j}^{post}(q - \xi_{r_j})\right] = \mathbb{E}_{\xi_{r_j}}\left[F_{r_j}^{post}(\tilde{q})\right],\ \forall \tilde{q} = q - \xi_{r_j}. \tag{13}$$

Lemma 3.1. *A lower bound on the expected recourse cost at the j^{th} customer for each q follows:*

$$\widetilde{F}_{r_j}(q) = \min_{v_{u_e} \in U_h^1} F_{r_j}(q)|_{r_j := u_e}, \tag{21}$$

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

$$\widetilde{F}_{r_{j-l+1}}(q) = \min_{U \subset U_h^1 : |U| = |U_h^1| - 1} \prod_{v_{u_e} \in U} p_{u_e}^{l^{min}} . \widetilde{F}_{r_j}(q), \tag{22}$$

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

Proof. By definition, $\min_{U \subset U_h^1 : |U| = |U_h^1| - 1} \prod_{v_{u_e} \in U} p_{u_e}^{l^{min}}$ 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 3.1), Equation (22) is a lower bound on the excepted recourse cost of the $j-l+1^{\text{th}}$. □

Using the bounds specified in Lemma 3.1 and Lemma 3.2, the recourse function (14) 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 (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 (14). 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 (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 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 4.1, while a comparison with the other policies is performed in Section 4.2.

We use the instances of Salavati-Khoshghalb et al. (2017), 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 1, 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 $\sum_{i=2,\ldots,n} c_{i1}/(n-1)$, which is the average distance to the depot when considering all customers. Furthermore, L is set to zero. For each combination in Table 1, ten instances were generated, thus yielding a total of 440 instances.

Table 1: Combinations of parameters to generate instances.

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

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 Gendron et al. (2005). The separation problem of constraints (4) is solved using the CVRPSEP package of Lysgaard et al. (2004). The maximum CPU time limit is set to 10 hours and the optimality gap was set to 0.01%.

4.1 Results for the hybrid recourse policy

The performance of the exact algorithm for the hybrid policy is presented in Tables 2-6 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. (2014) 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 2-6 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 6 .

The results in Tables 2-6 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. (2002) and Jabali et al. (2014). 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 2: 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 3: Hybrid 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: Hybrid 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 5: 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 6: 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		

4.2 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 7, 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 7: 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 8, 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 8: 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. (1995). 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 9 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 (2006) and Rei et al. (2010)). 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 8.

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 9: 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%

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.

Acknowledgment

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

Appendix

The L-shaped Algorithm

We briefly describe here the integer *L*-shaped algorithm (1). As a branch-and-cut algorithm, first, we state the initial current problem (CP) with relaxing the capacity / subtour-elimination constraints (4), and integrality constraints (7).

The integer *L*-shaped algorithm (1) 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 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) 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 (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**.

Acknowledgment

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

Algorithm 1 L-Shaped Algorithm

▷ state initial CP with the constraints: 2, 3, 5, 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^ν, Θ^ν) is the optimal solution of CP
▷ **Step 3**: check for any violation of (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

$$\sum_{\substack{1 \leq i \leq j \\ x_{ij}^\nu = 1}} x_{ij} \leq \sum_{1 \leq i \leq j} x_{ij}^\nu - 1 \tag{24}$$

end if

References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Lambert, V., G. Laporte, F.V. Louveaux. 1993. Designing collection routes through bank branches. *Computers & Operations Research* **20**(7) 783–791.

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

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

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

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

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

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

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

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

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

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

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

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