A Chance-Constrained Two-Echelon Vehicle Routing Problem with Stochastic Demands

N. Sluijk, A. M. Florio, J. Kinable, N.P. Dellaert, T. Van Woensel

March 2021

Abstract

Two-echelon distribution systems are often considered in city logistics to maintain economies of scale and satisfy the emission zone requirements in the cities. In this work, we formulate the two-echelon vehicle routing problem with stochastic demands as a chance-constrained stochastic optimization problem, where the total demand of the customers in each second-echelon route should fit within the vehicle capacity with a high probability. We propose two efficient solution procedures based on column generation. Key to the efficiency of these procedures is the underlying labeling algorithm to generate new columns. We propose a novel labeling algorithm based on simultaneous construction of second-echelon routes and a labeling algorithm that builds second-echelon routes sequentially. To further enhance the performance of the solution procedure, we use statistical inference techniques to ensure that the chance constraints are met. We reduce the number of customer combinations for which the chance constraint needs to be verified by imposing feasibility bounds on the stochastic customer demands. With these bounds, the runtimes of the labeling algorithms are reduced significantly. Finally, we show the value of the stochastic formulation in terms of improved solution cost and guaranteed feasibility of second-echelon routes.

1 Introduction

The retail e-commerce sales worldwide amounted to 3.53 trillion US dollars in 2019 and are projected to grow to 6.54 trillion US dollars in 2022 (Statista, 2020). At the same time, the last mile is becoming increasingly congested, and sustainability is receiving more attention (Agatz et al., 2008). An increasing number of European Union cities are enforcing low emission zones to ensure a less polluted environment and are preparing for a complete phase-out of internal combustion engines and/or zero-emission mobility areas (Lurkin et al., 2021; Transport Environment, 2018; TLN, 2021). Thus, there is a need for a sustainable distribution chain that satisfies the restricted emission zones of cities while maintaining economies of scale. This can be achieved by dividing the transportation chain into two echelons. Instead of directly transporting goods from depots to customers, large trucks deliver to satellite locations just outside of the cities. Here, goods are unloaded from the trucks and loaded into smaller vehicles that bring the goods to their final destinations. Such a distribution system is a practical example of a two-echelon vehicle routing problem (2E-VRP), where we operate large trucks on the first echelon to achieve economies of scale, and smaller vehicles on the second echelon to satisfy the emission zone requirements, e.g., with low emission vehicles or electric vehicles with limited range.

The deterministic 2E-VRP is a well-studied problem (Cuda et al., 2015). In reality, however, various parameters of the problem may be uncertain, e.g., travel times, customer demands, and/or customer presence. Although many papers consider stochastic variants of the classical single-echelon vehicle routing problem (Oyola et al., 2018), only two papers deal with stochastic variants of the 2E-VRP. Both Liu et al. (2017) and Wang et al. (2017a) consider stochastic customer demands and capture the uncertainty with a recourse action, in which the vehicle performs a replenishment trip to a satellite and/or depot whenever a customer's demand exceeds the remaining vehicle capacity.

In this paper, we consider the two-echelon vehicle routing problem with stochastic demands (2E-VRPSD). Instead of assuming a restocking policy, we capture demand stochasticity with a probabilistic capacity constraint on the second-echelon vehicle capacity. We require with high probability that the total demand of customers in a second-echelon route does not exceed the second-echelon vehicle capacity. In this way, we reduce the need for inventory coordination at satellites and potentially expensive replenishment trips from customers to the depot over a two-echelon network.

In summary, this paper brings the following contributions:

- We formulate the 2E-VRPSD as a chance-constrained stochastic optimization problem and propose two efficient solution procedures based on column generation. Key to the efficiency of these procedures is the underlying labeling algorithms to generate new columns.
- Specifically, we propose a novel labeling algorithm based on simultaneous construction of second-echelon routes and a labeling algorithm that builds second-echelon routes sequentially. To further enhance the performance of the solution procedure, we use statistical inference techniques to verify that the probabilistic capacity constraint is satisfied.
- Additionally, feasibility bounds on the stochastic customer demands are imposed to reduce the number of customer combinations for which the chance constraint needs to be verified. Our experiments show that with these bounds the labeling algorithm's runtimes are reduced by a factor of up to 1.9.
- Finally, a set of results shows the value of working with the stochastic demand formulation in terms of improved solution cost and guaranteed feasibility of second-echelon routes.

The remainder of this paper is organized as follows. In Section 2, the relevant literature is discussed, followed by the problem definition in Section 3. In Section 4, we propose two labeling algorithms to solve the column generation subproblem. In Section 5, we introduce several methods for evaluating the chance constraints and the notion of feasibility bounds. In Section 6, we present and discuss the results obtained on instances with independent and correlated demand distributions. Finally, in Section 7, we draw concluding remarks and provide perspectives for future research.

2 Literature Review

The two-echelon vehicle routing problem with stochastic demands is a combination of the vehicle routing problem with stochastic demands and the two-echelon vehicle routing problem. In the following, we discuss the relevant literature of both domains.

2.1 Vehicle Routing Problems with Stochastic Demands

The vehicle routing problem with stochastic demands (VRPSD) is the most studied variant of stochastic vehicle routing problems (Oyola et al., 2017). The stochasticity is often captured with a recourse policy or chance constraints. When considering recourse policies, the problem is formulated as a two-stage stochastic program and the aim is to minimize the first stage routing costs and second stage expected recourse costs. Three main recourse policies are frequently considered. With the *detour-to-depot* policy, also referred to as the classical recourse policy, the vehicle returns to the depot to resupply if it runs out of capacity and continues the planned route from the last visited customer (Dror et al., 1989). The *optimal restocking* policy allows preventive returns to the depot if a failure at a later customer is likely to occur and the expected cost of replenishing at the current customer is less than the expected cost of replenishing at a later customer (Salavati-Khoshghalb et al., 2019a; Florio et al., 2020; Louveaux and Salazar-González, 2018). Motivated by the use of fixed operational rules in practice, Salavati-Khoshghalb et al. (2019b) propose a rule-based policy, where the optimal restocking policy is replaced by some preset rules that establish customer-specific thresholds according to which the preventive return trips are governed. Finally, the *reoptimization* policy reoptimizes the route after the current customer demand is revealed, including possible replenishment trips (Secomandi and Margot, 2009).

Stochasticity in customer demand can also be captured with chance constraints. Chance constraints ensure that the probability of the customers' total demand not exceeding the vehicle capacity is above a certain threshold. Noorizadegan and Chen (2018) develop a branch-and-price algorithm to solve the VRPSD with chance constraints. They assume independent Poisson distributed demands and provide dominance rules based on distribution parameters that can be applied to independent demand distributions with the additive property. Dinh et al. (2018) replace the independence assumption by the condition that it must be possible to compute a quantile of the sum of the customer demands in a subset of customers and solve the model with a branch-cut-and-price algorithm.

An overview of solution methods for stochastic vehicle routing problems can be found in Oyola et al. (2018). Most of them are only applicable to instances with independent demand distributions. In practice, customer demands may be correlated, given the many factors that could induce correlation, including weather, events, and sales (Gendreau et al., 2016). The methodology we propose in this paper can also be used to handle realistic situations where customer demand is modeled with dependent, correlated, or data-driven distributions.

2.2 Two-Echelon Vehicle Routing Problems

The first application of a 2E-VRP was introduced by Jacobsen and Madsen (1980). Nearly two decades later, a formal description of the problem appeared in Crainic et al. (2004). Different variations of the 2E-VRP have been considered, including the 2E-VRP with time windows (Dellaert et al., 2019); electric vehicles (Breunig et al., 2019); covering options (Enthoven et al., 2020); swap containers (Mühlbauer and Fontaine, 2021); delivery options (Zhou et al., 2018); simultaneous pickup and delivery (Belgin et al., 2018); multiple commodities (Dellaert et al., 2021); grey zone customers (Anderluh et al., 2021); and satellite synchronization and multiple trips (Grangier et al., 2016). An extensive overview of various 2E-VRP applications as well as common solution approaches is provided in Cuda et al. (2015).

Recently, Marques et al. (2020) proposed a branch-cut-and-price algorithm for the single-depot 2E-VRP, including a route-based formulation that does not require explicit variables to cover the product flow at satellites, a new family of valid inequalities, and a new branching strategy on the number of first-echelon vehicles visiting a subset of satellites. Currently, this is the best exact solution method for the 2E-VRP.

Dellaert et al. (2019) consider the 2E-VRP with time windows and multiple depots and propose two path-based formulations. In the first formulation, paths are defined over both first- and second-echelon routes, whereas in the second formulation, the first- and second-echelon paths are decomposed. Branch-and-price-based algorithms are developed for both formulations. New instances are proposed for the setting with multiple depots and time windows. Instances with up to 100 customers and five satellites are solved to optimality.

Heuristic approaches have been developed to obtain acceptable solutions in a short amount of time, including math-based heuristics (Perboli et al., 2011), large neighborhood search (Kitjacharoenchai et al., 2020; Wang et al., 2017b), adaptive large neighborhood search (Breunig et al., 2016; Grangier et al., 2016; Jie et al., 2019; Li et al., 2020; Enthoven et al., 2020), variable neighborhood descent (Belgin et al., 2018), greedy randomized adaptive search procedure (Zeng et al., 2014; Anderluh et al., 2017), and population-based heuristics (Zhou et al., 2018).

While the literature on the deterministic 2E-VRP is vast, only two papers consider the 2E-VRPSD. Liu et al. (2017) design a simulation-based tabu-search algorithm to solve the 2E-VRPSD. Given the two-echelon structure, the removal and insertion of both satellites and customers are considered. The authors consider two attribute sets for the tabu list, one containing combinations of satellites and first-echelon vehicles and the other containing combinations of customers, satellites, and second-echelon vehicles. The cost of a neighborhood move is estimated with Monte Carlo simulation.

Wang et al. (2017a) develop a genetic algorithm to solve the 2E-VRPSD, where a chromosome represents the second-echelon routes that are part of a complete solution. A decoding method is proposed to translate a chromosome to a complete solution, including first-echelon routes. Mutation operators related to two decisions are incorporated: the selection of satellites and second-echelon routes. Exact methods and discrete approximation methods are considered for calculating the expected route failure cost.

Thus, so far, only heuristic methods have been considered for the 2E-VRPSD. As well known, a drawback of heuristics is that they do not guarantee the optimality of the solution. In this paper, we solve the 2E-VRPSD with a column generation-based algorithm, which provides us with lower bounds on the solution values, thereby enabling us to compute optimality gaps. As opposed to considering a recourse policy and requiring independent demand distributions as in Wang et al. (2017a) and Liu et al. (2017), we capture the stochasticity with chance constraints and do not pose any assumption on the demand distributions.

3 Problem Definition

The chance-constrained 2E-VRPSD is defined on an undirected weighted graph $G = (V, E)$ with vertex set $V = D \cup S \cup C$, representing the set of depots, satellites and customers, respectively. The edge set E is partitioned into two sets, one for each echelon. Set $E^1 = \{(i,j)|i \in D \cup S, j \in S, i \neq j\}$ consists of the edges that can be traversed by first-echelon (FE) vehicles. Similarly, set $E^2 = \{(i,j)|i \in S \cup C, j \in C, i \neq j\}$ contains the connections that can be used by the second-echelon (SE) vehicles. The weight c_e of edge $e \in E$ is equal to the transportation cost (e.g. distance) of traversing edge e. For each echelon we have an unlimited homogeneous fleet of vehicles. The FE and SE vehicles have a capacity of Q^{FE} and Q^{SE}, respectively, where $Q^{\text{FE}} > Q^{\text{SE}}$.

At the satellites, goods are unloaded from the FE vehicles and directly loaded into the SE vehicles, without any intermediate storage. To avoid the complex synchronization needed when an SE vehicle has to wait for multiple FE vehicles, we do not allow deliveries from different FE vehicles to the same SE vehicle. A satellite can be visited by multiple FE vehicles, as long as they supply different SE vehicles. A formal definition of an FE route is given in Definition 1.

Definition 1 (First-echelon route)**.** *A first-echelon route is a sequence* $f = (n_0, n_1, \ldots, n_m, n_{m+1})$ *where* $m \geq 1$, $n_0 = n_{m+1} \in D$, $n_i \in S$ *for all* $i \in \{1, \ldots, m\}$, *and* $n_i \neq n_j$ *for all* $i, j \in \{1, \ldots, m\}$, $i \neq j$.

Analogous to an FE route, an SE route starts and ends at a satellite and visits one or more customers. The demand ξ_i of each customer $i \in C$ is a non-negative integer random variable with mean μ_i and variance σ_i^2. Probabilistic capacity constraints are imposed on the SE routes. An SE route is considered to be *capacity-feasible* if the total demand of the customers visited in the route does not exceed the vehicle capacity Q^{SE} with a high probability of at least η. We formalize as follows:

Definition 2 (Second-echelon route)**.** *A valid second-echelon route is a sequence* $r = (n_0, n_1, \ldots, n_m, n_{m+1})$, *where* $m \geq 1$, $n_0 = n_{m+1} \in S$, $n_i \in C$, $n_i \neq n_j$ *for all* $i, j \in \{1, \ldots, m\}$, $i \neq j$, *and*

$$\mathbb{P}\left(\sum_{i \in C'} \xi_i \leq Q^{SE}\right) \geq \eta, \tag{1}$$

with $C' = \{n_1, \ldots, n_m\}$.

Without loss of generality, let $\bar{Q}$ be a value such that Constraint (1) does not hold for any customer combination $C' \subset C$ with $\sum_{i \in C'} \mu_i > \bar{Q}$. In Section 5.3, the process of obtaining $\bar{Q}$ is described. We do not place any restrictions on the demand distributions. In particular, the demand of different customers can be statistically dependent.

To formally define the chance-constrained 2E-VRPSD through a mathematical model, similar to Dellaert et al. (2019), we introduce the notion of a *tour-tree*. A tour-tree is composed of a single FE route and all SE routes supplied by the FE route.

Definition 3 (Tour-tree)**.** *A tour-tree is a sequence* $t = (n_0, (n_1, R_1), \ldots, (n_m, R_m), n_{m+1})$, *with* $m \geq 1$, $n_0 = n_{m+1} \in D$, $n_k \in S$ *for* $k \in \{1, \ldots, m\}$, *and* R_k *representing the set of SE routes starting at satellite* n_k.

From Definition 3 it follows that any solution to the 2E-VRPSD can be expressed in terms of a finite set of tour-trees. The objective of the 2E-VRPSD is to find a set of tour-trees of minimum total

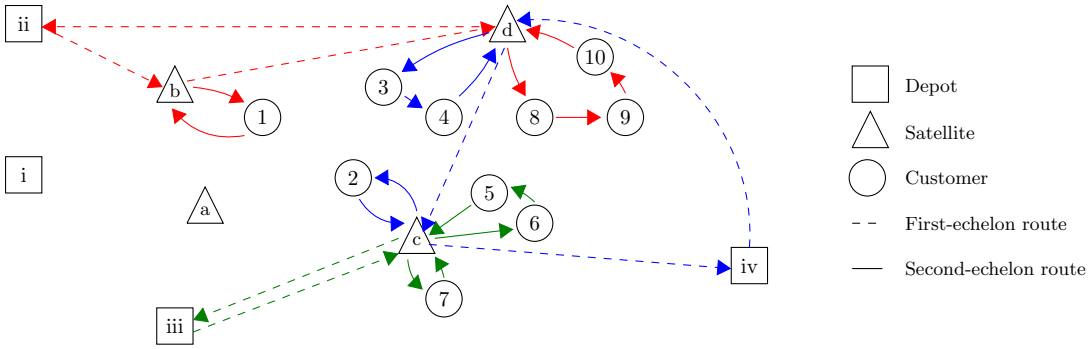

Figure 1: 2E-VRPSD instance and a potential solution

cost such that all customers are visited and the vehicle capacities are satisfied with high probability. Figure 1 shows an example of a solution for an 2E-VRPSD instance consisting of the following tour-trees: (ii,(b,(b,1,b)),(d,(d,8,9,10,d)),ii), (iii,(c,((c,7,c), (c,6,5,c))),iii), and (iv,(d,(d,3,4,d)),(c,(c,2,c)),iv).

We assume that each SE route leaves the satellite with full capacity, therefore an FE route can supply at most $\theta = \left\lfloor \frac{Q^{\mathrm{FE}}}{Q^{\mathrm{SE}}} \right\rfloor$ SE routes. We denote the set of edges that are traversed in tour-tree t by $E(t)$ and define its corresponding transportation costs as $c_t = \sum_{e \in E(t)} c_e$. Let T denote the set of all feasible tour-trees and a_{it} be a binary parameter equal to 1 if customer $i \in C$ is visited in tour-tree t. Using binary variables $x_t, t \in T$, the 2E-VRPSD can now be formulated as the following set-partitioning problem:

$$\begin{aligned} \text{(MP)} \quad \text{minimize} \quad & \sum_{t \in T} c_t x_t, \\ \text{subject to} \quad & \sum_{t \in T} a_{it} x_t = 1, \qquad i \in C, \\ & x_t \in \{0,1\}, \quad t \in T. \end{aligned} \tag{2}$$

The objective function minimizes the total cost of the selected tour-trees. Constraints (2) specify that each customer $i \in C$ is covered by exactly one tour-tree.

Problem MP assumes a set of feasible tour-trees, which grows exponentially with the number of customers in the problem instance, and cannot be solved directly for most 2E-VRPSD instances. Instead, we consider the linear relaxation of MP, hereafter referred to as RMP, and solve it with column generation.

To obtain tighter lower bounds, we impose the following bound on the minimum number of tour-trees in any feasible 2E-VRPSD solution:

$$\sum_{t \in T} x_t \geq \left\lceil \frac{\sum_{i \in C} \mu_c}{\theta \bar{Q}} \right\rceil. \tag{3}$$

4 Column Generation for the 2E-VRPSD

The column generation procedure iteratively solves a restricted version of RMP, in which the set of tour-trees T is replaced by a limited subset $T' \subset T$. At each iteration, a pricing problem is solved to generate new columns (tour-trees) with negative reduced cost, which are consecutively added to T'. The procedure terminates when no more columns with negative reduced cost can be identified.

The remainder of this section is structured as follows. In Section 4.1, we formally define the pricing

problem. In Sections 4.2 and 4.3, we propose two labeling algorithms for solving the pricing problem. The first labeling algorithm is based on simultaneous labeling of SE routes, whereas the second algorithm constructs SE routes sequentially. Finally, in Section 4.4, we discuss the use of completion bounds to control the growth of the labels.

4.1 Pricing Problem

To formally define the pricing problem, let λ_i, $i \in C$, and δ be the dual values corresponding to Constraints (2) and (3), respectively. The pricing problem can be formulated as a variant of the elementary shortest path problem with resource constraints (ESPPRC; Feillet et al., 2004) with capacity constraints for both FE and SE vehicles. The reduced cost c'_t of a tour-tree t is given by:

$$c'_t = c_t - \sum_{i \in C} a_{it}\lambda_i - \delta.$$

The objective of the pricing problem is to find a tour-tree t such that $c'_t < 0$. The ESPPRC is often solved with a labeling algorithm, where labels represent partial paths in graph G. The efficiency of a labeling algorithm depends on its ability to prune non-promising paths. Two techniques that are often considered to control the growth of labels are completion bounds and dominance rules. A completion bound is a lower bound on the reduced cost of all routes that can be obtained as extensions of the current label (Costa et al., 2019). Dominance rules are imposed on the available resources of a label to detect and remove redundant labels. One resource that is often used in the dominance rules is the remaining vehicle capacity. However, due to the stochastic nature of the customer demands, it is no longer possible to use this resource for controlling the growth of the labels. Additionally, efficient approaches for verifying the chance constraint are required. In the following, we focus on developing methodology that addresses these challenges and efficiently solves the RMP. Once an optimal solution to the RMP is obtained, we use the resulting tour-trees in T' to compute an upper bound to MP by solving problem MP directly for T' instead of T.

4.2 Multi-label Algorithm

Each label in the multi-label algorithm represents a partial tour-tree and consists of multiple partial SE routes, hence the name *multi-label*. With this structure, it is possible to start a new SE route while another SE route in the same tour-tree is still being formed. In contrast, existing labeling algorithms for constructing a tour-tree for the 2E-VRP consider SE routes consecutively, i.e., a new SE route can only be started after the previous SE route has been completed. The multi-label approach's main benefit is that we can enforce completion bounds on the remaining SE vehicle capacities rather than the remaining FE vehicle capacity. This enables us to implement the multi-label algorithm without dominance rules, making it fit for dependent, correlated, or data-driven demand distributions.

The multi-label algorithm starts with generating all possible permutations of the satellites that could be visited in an FE route. Each satellite represents the start of a different SE route. As stated before, we assume that each FE route can supply at most θ SE routes, and, as such, each permutation consists of at most θ satellites. Moreover, following Baldacci et al. (2013), we assume that the number of satellites $|S|$ is limited. For reasonable values of $|S|$ and θ, we can enumerate all possible $|S|^\theta$ permutations. We refer to a permutation as a *configuration*:

Table 1: Attributes of a Multi-label $\mathscr{L}$

Attribute		Description
Π		Configuration
ϕ		Total reduced cost
Ψ		Set of customers visited
π_k	$k \in \{1, \ldots, \lvert\Pi\rvert\}$	Starting satellite of path k
n_k,	$k \in \{1, \ldots, \lvert\Pi\rvert\}$	Last node visited by path k
ϕ_k^{SE},	$k \in \{1, \ldots, \lvert\Pi\rvert\}$	Total reduced cost of path k
$\bar{\mu}_k$,	$k \in \{1, \ldots, \lvert\Pi\rvert\}$	Sum of the customer mean demands in path k
$\bar{\sigma}_k^2$,	$k \in \{1, \ldots, \lvert\Pi\rvert\}$	Sum of the customer demand variances in path k
m_k,	$k \in \{1, \ldots, \lvert\Pi\rvert\}$	Number of customers visited in path k

Definition 4 (Configuration)**.** *A configuration is a sequence* $\Pi = (n_1, n_2, \ldots, n_k)$ *where* $n_i \in S$ *for all* $i \in \{1, \ldots, k\}$ *with* $k \leq \theta$*.*

The main differences between the definitions of a configuration and a tour-tree are the absence of depots and the satellites' multiplicity. For example, the configurations of the tour-trees depicted in Figure 1 are $\{b, d\}$, $\{c, c\}$, and $\{d, c\}$.

For each configuration, we use the depot that minimizes the total FE routing costs. Next, we create a multi-label for each configuration. The attributes of a multi-label $\mathscr{L}$ are given in Table 1, where we have general attributes related to the tour-tree and specific attributes for each partial SE route (path) $\mathscr{L}_k$, $k \in \{1, \ldots, |\mathscr{L}(\Pi)|\}$. Next to the common attributes of a label, we also store attributes $\bar{\mu}_k$, $\bar{\sigma}_k^2$, and m_k, $k \in \{1, \ldots, |\mathscr{L}(\Pi)|\}$, to expedite the process of checking the chance constraint and adherence to the symmetry breaking rules, which are further detailed in Section 4.2.

In each iteration of the multi-label algorithm, we randomly select the next multi-label $\mathscr{L}$ to be extended. For each path $\mathscr{L}_k$, $k \in \{1, \ldots, |\mathscr{L}(\Pi)|\}$, we iteratively select a customer $i \in C$ and check whether the extension of path $\mathscr{L}_k$ to customer i is feasible. An extension is feasible if the customer is not yet visited, it does not present a case of symmetry that is forbidden by the symmetry breaking rules, and the probabilistic capacity constraint is satisfied. The verification of the probabilistic capacity constraint is discussed in Section 5. If feasible, path $\mathscr{L}_k$ is extended to customer i, resulting in a new multi-label $\mathscr{L}'$. To obtain this multi-label, we copy all attributes of multi-label $\mathscr{L}$, since only the general attributes and the attributes related to path $\mathscr{L}'_k$ require updating:

$$
\begin{aligned}
\mathscr{L}' &= \mathscr{L}, & \mathscr{L}'(\phi_k^{\text{SE}}) &= \mathscr{L}(\phi_k^{\text{SE}}) + c_{\mathscr{L}(n_k),i} - \lambda_i, \\
\mathscr{L}'(\phi) &= \mathscr{L}(\phi) + c_{\mathscr{L}(n_k),i} - \lambda_i, & \mathscr{L}'(\bar{\mu}_k) &= \mathscr{L}(\bar{\mu}_k) + \mu_i, \\
\mathscr{L}'(\Psi) &= \mathscr{L}(\Psi) \cup \{i\}, & \mathscr{L}'(\bar{\sigma}_k^2) &= \mathscr{L}(\bar{\sigma}_k^2) + \sigma_i^2, \\
\mathscr{L}'(n_k) &= i, & \mathscr{L}'(m_k) &= \mathscr{L}(m_k) + 1.
\end{aligned}
$$

After extending multi-label $\mathscr{L}$ to $\mathscr{L}'$, we transform the partial tour-tree represented by $\mathscr{L}'$ into a complete tour-tree, by returning each path to its starting satellite, and evaluate the corresponding total reduced cost:

$$
\phi_t = \mathscr{L}'(\phi) + \sum_{k=1}^{|\mathscr{L}'(\Pi)|} c_{\mathscr{L}'(n_k),\mathscr{L}'(\pi_k)} - \delta. \tag{4}
$$

If the total reduced cost is negative, the corresponding tour-tree is added to the master problem.

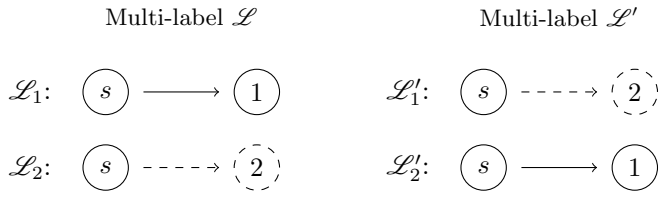

Figure 2: Two multi-labels containing the same information but in opposite paths.

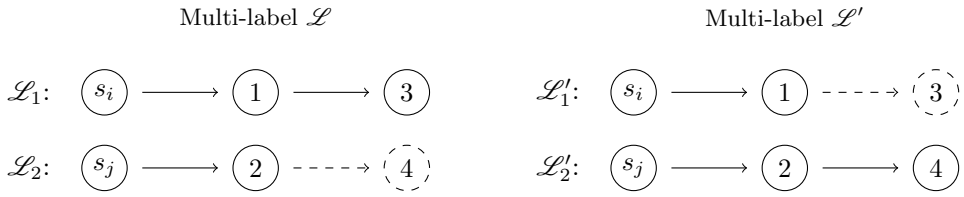

Figure 3: Two multi-labels with paths starting from, possibly different, satellites s_i and s_j, resulting in the same multi-label when the dashed extension is performed.

Symmetry breaking rules.

The multi-label algorithm, when left untreated, is likely to produce a large number of symmetrical tour-trees, e.g. a pair of tour-trees with mirrored SE routes. To reduce the occurrences of symmetry in the multi-label algorithm, we incorporate several symmetry breaking rules. In the following, we provide examples of three types of symmetry. After each example, we introduce a rule to eliminate the possibility of this symmetry occurring. For notational convenience, we define $\alpha(\mathscr{L}_k)$ and $\zeta(\mathscr{L}_k)$ to be the first and last customer visited in path $\mathscr{L}_k$, respectively, for $k \in \{1, \ldots, |\mathscr{L}(\Pi)|\}$. To express the various symmetry breaking rules, we impose some arbitrary ordering on the elements of C, e.g., by assigning each customer a unique number. We write $i \prec j$ to denote that customer i precedes customer j in this ordering.

The first example is given in Figure 2, where two multi-labels, with paths starting from the same satellite, are given. In the first iteration of the multi-label algorithm, customer 1 is added to $\mathscr{L}_1$ and $\mathscr{L}'_2$. In the second iteration, customer 2 is added to $\mathscr{L}_2$ and $\mathscr{L}'_1$. This results in two multi-labels containing the same information but in opposite paths. This case of symmetry can be avoided by enforcing that the first customers of paths starting from the same satellite should be in increasing order. Formally:

Rule 1. *If $\mathscr{L}(\pi_k) = \mathscr{L}(\pi_{k+1})$, then $\alpha(\mathscr{L}_k) \prec \alpha(\mathscr{L}_{k+1})$ must hold, for all $k \in \{1, \ldots, |\mathscr{L}(\Pi)| - 1\}$.*

Symmetry can also be present when adding the kth customer to a path, with $k > 1$. For example, in Figure 3, two multi-labels with paths starting from, possibly different, satellites s_i and s_j are given. Before performing the extension indicated by the dashed line, the multi-labels represent different tour-trees. However, identical multi-labels are obtained when adding customer 4 to $\mathscr{L}_2$ and customer 3 to $\mathscr{L}'_1$.

To prevent this symmetry in the multi-labels, we only allow the extension of a path if all subsequent paths visit at most one customer. Formally:

Rule 2. *Path $\mathscr{L}_k$ is only extended if paths $\mathscr{L}_j$ visit at most one customer, i.e. $\mathscr{L}(m_j) \leq 1$, for all $j \in \{k+1, \ldots, |\mathscr{L}(\Pi)|\}$.*

This rule does not eliminate the possibility of having multi-labels with paths visiting multiple customers. It only restricts the order in which they are being created, i.e. we obtain a multi-label with paths visiting

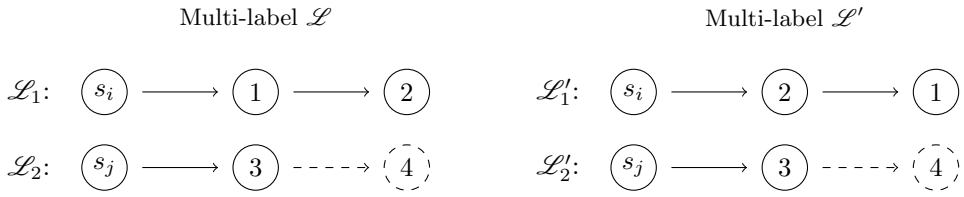

Figure 4: Two multi-labels containing the same customers in the first paths, but in opposite order.

multiple customers by first extending the first path to multiple customers, then the second path, etc. Once we have extended the second path, Rule 2 does not allow further extension of the first path. However, there will be other multi-labels with the desired extension of the first path. For example, in Figure 3, we cannot perform the extension indicated by the dashed line of multi-label $\mathscr{L}'$, but the desired multi-label can be obtained from multi-label $\mathscr{L}$.

The last symmetry rule concerns the orientation of customers in paths that can no longer be extended. In Figure 4, paths $\mathscr{L}_1$ and $\mathscr{L}'_1$ can no longer be extended once the dashed extensions of paths $\mathscr{L}_2$ and $\mathscr{L}'_2$ are performed. Both $\mathscr{L}_1$ and $\mathscr{L}'_1$ visit the same sequence of customers, but in the opposite order, i.e., they represent the same SE route.

In general, if a path $(s, n_0, \ldots, n_m)$, with $s \in S$, $m > 0$ and $n_i \in C$ for $i \in \{0, \ldots, m\}$, is feasible, then the reverse path, $(s, n_m, \ldots, n_0)$, is also feasible, therefore both paths will be constructed. To eliminate this symmetry, we only allow the addition of the k-th customer, $k > 1$, to a path if for all preceding paths it holds that the first and last customers are visited in increasing order. Formally:

Rule 3. *Path $\mathscr{L}_k$ is only extended if $\alpha(\mathscr{L}_j) \prec \zeta(\mathscr{L}_j)$ for all $j \in \{1, \ldots, k-1\}$.*

4.3 Two-Echelon One-Path Algorithm

To establish a benchmark for the multi-label algorithm, we develop an alternative pricing algorithm where the SE routes are built sequentially. We refer to this alternative approach as the Two-Echelon One Path (2E-1P) algorithm. The 2E-1P algorithm is inspired by one of the algorithms proposed by Dellaert et al. (2019). In contrast to the multi-label algorithm (Section 4.2), which generates elementary paths where customers can only be visited once, in the 2E-1P algorithm we relax this requirement and allow *ng*-paths (Baldacci et al., 2011) instead. In an *ng*-path, customer $i \in C$ can be revisited if, between this visit and the previous visit to customer i, a customer $j \in C$ is visited for which it holds that i is not in the neighborhood N_j of j, where N_j consists of the k nearest customers to customer j, $0 \leq k \leq |C|$. In this way, cycles contain customers that are far away from each other. The customers to which the current label cannot be extended without violating the *ng*-restriction are stored in an *ng*-set. An overview of the attributes of a label $\mathfrak{L}$ in the 2E-1P algorithm is given in Table 2.

In each iteration of the 2E-1P algorithm, we randomly select the next label $\mathfrak{L}$ to be extended and iteratively consider a node $i \in V$ for extension. Let F_j denote the cumulative density function (CDF) of customer demand ξ_j, $j \in C$, and $(\mathfrak{L}(F) * F_j)$ the CDF that we obtain after convoluting $\mathfrak{L}(F)$ with F_j. An extension is feasible if one of the following three conditions holds:

$$(1) \qquad i \in C \wedge (\mathfrak{L}(F) * F_i)(Q^{\mathrm{SE}}) \geq \eta \wedge i \notin \mathfrak{L}(\Gamma) \wedge \{\mathfrak{L}(n) \in C \vee \{\mathfrak{L}(n) \in S \wedge \mathfrak{L}(p) \in S \cup D\}\},$$

$$(2) \qquad i \in S \wedge \{\mathfrak{L}(n) \in D \vee \{\mathfrak{L}(n) \in C \wedge \mathfrak{L}(s) = i\} \vee \{\mathfrak{L}(n) \in S \wedge \mathfrak{L}(p) \in C \wedge i \in \mathfrak{L}(U)\}\},$$

Table 2: Components of Label $\mathfrak{L}$ in the 2E-1P Algorithm

Notation	Description
d	Starting depot of the tour-tree
s	Last visited satellite
n	Last visited node
p	Second to last visited node
Γ	ng-set
U	Set of usable satellites
ϕ	Total reduced cost
r	Number of finished SE routes
$\bar{\mu}$	Sum of customer mean demands in current SE route
$\bar{\sigma}^2$	Sum of customer demand variances in current SE route
F	Cumulative joint density function of the customer demands in current SE route

$$(3) \qquad i \in D \wedge \mathfrak{L}(d) = i \wedge \{\mathfrak{L}(n) \in S \wedge \mathfrak{L}(p) \in C\}.$$

If feasible, label $\mathfrak{L}$ is extended to node i, resulting in a new label $\mathfrak{L}'$. The updated attributes of label $\mathfrak{L}'$ are obtained as follows:

$$\mathfrak{L}'(d) = \mathfrak{L}(d),$$

$$\mathfrak{L}'(s) = \begin{cases} i, & \text{if } i \in S, \\ \mathfrak{L}(s), & \text{otherwise,} \end{cases}$$

$$\mathfrak{L}'(n) = i,$$

$$\mathfrak{L}'(p) = \mathfrak{L}(n),$$

$$\mathfrak{L}'(\Gamma) = \begin{cases} (\mathfrak{L}(\Gamma) \cap N_i) \cup \{i\}, & \text{if } i \in C, \\ \mathfrak{L}(\Gamma), & \text{otherwise,} \end{cases}$$

$$\mathfrak{L}'(U) = \begin{cases} \mathfrak{L}(U) \setminus \{\mathfrak{L}(s)\}, & \text{if } i \in S, \mathfrak{L}(s) \neq i, \\ \mathfrak{L}(U), & \text{otherwise,} \end{cases}$$

$$\mathfrak{L}'(\phi) = \begin{cases} \mathfrak{L}(\phi) + c_{\mathfrak{L}(n),i} - \lambda_i, & \text{if } i \in C, \\ \mathfrak{L}(\phi) + c_{\mathfrak{L}(n),i}, & \text{otherwise,} \end{cases}$$

$$\mathfrak{L}'(r) = \begin{cases} \mathfrak{L}(r) + 1, & \text{if } i \in S, \mathfrak{L}(n) \in C, \\ \mathfrak{L}(r), & \text{otherwise,} \end{cases}$$

$$\mathfrak{L}'(\bar{\mu}) = \begin{cases} \mathfrak{L}(\bar{\mu}) + \mu_i, & \text{if } i \in C, \\ 0, & \text{otherwise,} \end{cases}$$

$$\mathfrak{L}'(\bar{\sigma}^2) = \begin{cases} \mathfrak{L}(\bar{\sigma}^2) + \sigma_i^2, & \text{if } i \in C, \\ 0, & \text{otherwise,} \end{cases}$$

$$\mathfrak{L}'(F) = \begin{cases} \text{undefined}, & \text{if } i \in D \cup S, \\ F_i, & \text{if } i \in C \text{ and } \mathfrak{L}(n) \in S, \\ \mathfrak{L}(F) * F_i, & \text{otherwise.} \end{cases}$$

Note that the convolution of two CDFs is only possible with independent demand distributions. As a result, the 2E-1P algorithm can only be applied to instances with independent demand distributions.

To reduce the number of labels created in each iteration of the labeling procedure, we apply the following dominance rule:

Definition 5 (Exact dominance)**.** *Label* $\mathfrak{L}_1$ *dominates label* $\mathfrak{L}_2$ *if and only if*

(i) $\mathfrak{L}_1(d) = \mathfrak{L}_2(d)$,

(ii) $\mathfrak{L}_1(s) = \mathfrak{L}_2(s)$,

(iii) $\mathfrak{L}_1(n) = \mathfrak{L}_2(n)$,

(iv) $\mathfrak{L}_1(\Gamma) \subseteq \mathfrak{L}_2(\Gamma)$,

(v) $\mathfrak{L}_2(U) \subseteq \mathfrak{L}_1(U)$,

(vi) $\mathfrak{L}_1(\phi) \leq \mathfrak{L}_2(\phi)$,

(vii) $\mathfrak{L}_1(r) \leq \mathfrak{L}_2(r)$,

(viii) $\mathfrak{L}_1(F(x)) \leq \mathfrak{L}_2(F(x)) \qquad \forall x \in \{0, \ldots, Q^{SE}\}$,

where one of the requirements (iv)-(viii) has to hold strictly.

Definition 5 states that if $\mathfrak{L}_1$ and $\mathfrak{L}_2$ represent two partial tour-trees with the same last visited node, extending $\mathfrak{L}_1$ should always result in a tour-tree with lower or equal reduced cost. Conditions (i) - (iii) ensure that both labels start from the same depot and have the same last visited satellite and customer. Condition (iv) requires that the set of customers to which $\mathfrak{L}_1$ cannot be extended is a subset of the set of customers to which $\mathfrak{L}_2$ cannot be extended. Condition (v) ensures that the set of satellites that can be visited from $\mathfrak{L}_1$ contains the satellites that can be visited from $\mathfrak{L}_2$. Condition (vi) assures that $\mathfrak{L}_1$ has a smaller or equal reduced cost than $\mathfrak{L}_2$. Condition (vii) requires that $\mathfrak{L}_1$ contains at most the same number of SE routes as $\mathfrak{L}_2$, which, given the assumption that SE vehicles leave the satellites with full capacity, is a sufficient condition on the FE vehicle capacity. Finally, Condition (viii) ensures that the CDF of $\mathfrak{L}_1$ has a lower or equal value than the CDF of $\mathfrak{L}_2$ for all values of $x \in \{0, \dots, Q^{\text{SE}}\}$.

Comparing CDFs for all $x \in \{0, \dots, Q^{\text{SE}}\}$ is a costly operation. Therefore, we also consider a heuristic dominance rule, where instead of enforcing Condition (viii), we enforce Conditions (ix)-(x). Formally:

Definition 6 (Heuristic dominance)**.** *Label* $\mathfrak{L}_1$ *dominates label* $\mathfrak{L}_2$ *if*

$$\textit{(i) - (vii)},$$

$$\textit{(ix)} \quad \mathfrak{L}_1(\bar{\mu}) \leq \mathfrak{L}_2(\bar{\mu}),$$

$$\textit{(x)} \quad \mathfrak{L}_1(\bar{\sigma}^2) \leq \mathfrak{L}_2(\bar{\sigma}^2),$$

where one of the requirements (iv) - (vii), (ix) - (x) has to hold strictly.

When searching for new columns, we first consider the heuristic dominance rule, which is easier to evaluate, but also more aggressive, than the exact dominance rule. As a consequence, it sometimes falsely claims that one label dominates another label. Therefore, if no columns with negative reduced cost are found when solving the pricing problem with the heuristic dominance rule in the 2E-1P algorithm, we solve the pricing problem again using the exact dominance rule in the labeling algorithm.

Acceleration technique for checking dominance.

To enable faster processing of the exact and heuristic dominance rules, we group the labels on the last node visited and *ng*-set. In each iteration of the labeling procedure, we start with processing dominance among the labels to be extended. First, we perform dominance checks *within* *ng*-sets, that is, we iterate over all labels with the same last node visited and the same *ng*-set and process dominance. Next, we check for dominance *among* *ng*-sets by comparing all labels with last node $n \in S \cup C$ and *ng*-set Γ to labels with last node n and *ng*-set Γ' such that $\Gamma' \subset \Gamma$.

4.4 Completion Bounds

In both pricing algorithms, we control the growth of the labels by computing a lower bound on the reduced cost that can be obtained when considering all possible extensions from a label (completion bound). If this bound is non-negative, the label can be safely discarded. We consider completion bounds based on the resource-constrained shortest path (RCSP; Feillet et al., 2004). Let $\psi(i, j, q)$ be the value of the resource-constrained shortest path in graph $G = (V, E)$ from node i to node j with a capacity of q, with G as defined

in Section 3. The cost of an edge $(i, j) \in E$ equals:

$$d_{ij} = \begin{cases} c_{ij} - \lambda_j, & \text{if } j \in C, \\ c_{ij}, & \text{otherwise,} \end{cases} \tag{5}$$

where λ_j, $j \in C$, are the dual values obtained from Constraint (2). Visiting a node $i \in C$ consumes μ_i units of q. In the following, we formulate the completion bounds for the labels in each of the pricing algorithms.

To derive the completion bound $\mathcal{CB}(\mathscr{L})$ on multi-label $\mathscr{L}$, we first compute the RCSP-bound for each path $\mathscr{L}_k$, $k \in \{1, \ldots, |\mathscr{L}(\Pi)|\}$. Next, we combine the RCSP-bounds of the paths in a multi-label to evaluate the completion bound on the multi-label. Specifically, in (4) we substitute the costs of finishing a path with its RCSP-bound to obtain the completion bound on $\mathscr{L}$, that is:

$$\mathcal{CB}(\mathscr{L}) = \mathscr{L}(\phi) + \sum_{k=1}^{|\mathscr{L}(\Pi)|} \psi\left(\mathscr{L}(n_k), \mathscr{L}(\pi_k), \bar{Q} - \mathscr{L}(\bar{\mu}_k)\right) - \delta,$$

where $\bar{Q}$ is used as the upper bound on the SE vehicle capacity.

Suppose now that path $\mathscr{L}_k$ has a positive completion bound, specifically,

$$\mathscr{L}(\phi_k) + \psi\left(\mathscr{L}(n_k), \mathscr{L}(\pi_k), \bar{Q} - \mathscr{L}(\bar{\mu}_k)\right) > 0,$$

for some $k \in \{1, \ldots, |\mathscr{L}(\Pi)|\}$. Creating a multi-label without path $\mathscr{L}_k$ would result in a multi-label with lower reduced cost. Therefore, if any of the paths have a positive completion bound, the multi-label is eliminated. In this way, non-promising multi-labels can be discarded at an early stage of the pricing algorithm.

The completion bound $\mathcal{CB}(\mathfrak{L})$ for a label $\mathfrak{L}$ in the 2E-1P algorithm can be derived in a similar way, using the remaining FE vehicle capacity rather than the remaining SE vehicle capacity. Through $\bar{Q}$ and the assumption that each FE route can supply at most θ SE routes, we derive that an FE vehicle can satisfy a total mean demand of at most $\theta\bar{Q}$. The completion bound on a label $\mathfrak{L}$ can now be computed as follows:

$$\mathcal{CB}(\mathfrak{L}) = \mathfrak{L}(\phi) + \psi\left(\mathfrak{L}(n), \mathfrak{L}(d), \theta \cdot \bar{Q} - \mathfrak{L}(r) \cdot \bar{Q} - \mathfrak{L}(\bar{\mu})\right) - \delta.$$

Each time the pricing algorithm is invoked, we pre-compute the RCSP-bounds for every $i \in C$, $j \in S$, and $q \in \{0, \ldots, \bar{Q}\}$ in case of the multi-label algorithm, and for every $i \in S \cup C$, $j \in D$ and $q \in \{0, \ldots, \theta\bar{Q}\}$ in case of the 2E-1P algorithm. In this way, the RCSP bounds can be retrieved in constant time during the labeling procedures.

5 Evaluating Probabilistic Capacity Constraints

Whenever we extend an SE route to another customer, we assess whether the resulting partial SE route satisfies the probabilistic capacity constraint (Constraint (1)). We consider two methods for verifying feasibility. The first one is by convolution of the probability distributions of the customers visited in an SE route and verifying feasibility from the convoluted distribution (Section 5.1). Alternatively, instead of computing this probability exactly, we could also approximate it with statistical inference tests (Section 5.2). The benefit of using statistical inference, over convoluting the distributions, is that (in)feasibility can be confirmed much faster if a customer combination is clearly (in)feasible. Nevertheless, even with statistical inference,

checking feasibility remains a costly operation, thus, from a computational point of view, it should be done as infrequent as possible. For this purpose, we introduce the notion of feasibility bounds in Section 5.3.

5.1 Convolutions of Probability Distributions

When the customer demands are independent random variables, their corresponding probability distributions can be convoluted to obtain a probability distribution of the sum of the original random variables. In general, if we have random non-negative integer variables X and Y, the distribution of the sum $Z = X + Y$ can be obtained by:

$$\mathbb{P}(Z = z) = \sum_{k=0}^{z} \mathbb{P}(x = k)\mathbb{P}(y = z - k).$$

Let $C' \subset C$ denote the set of customers in some (partial) SE route for which we want to verify feasibility. We convolute the demand distributions of the customers in C' iteratively, from 0 to Q^{SE}, since the focus is on computing the probability that the total demand of the customers does not exceed the SE vehicle capacity (feasibility probability). The complexity of this algorithm is $\mathcal{O}(|C'| \cdot (Q^{\text{SE}})^2)$, thus linear in the number of customers in C'.

5.2 Statistical Inference with Monte Carlo Sampling

In statistical inference, scenario generation is used to estimate the probability $\hat{\eta}$ that the total demand of the customers in an SE route does not exceed the SE vehicle capacity. The method consists of iteratively generating a set of scenarios and computing a confidence interval $[a, b]$ around the feasibility probability, i.e., $a \leq \hat{\eta} \leq b$ (Florio et al., 2021). We use the Agresti-Coull interval to estimate this interval (Brown et al., 2001).

We consider five standard deviations when defining the interval, which translates to a probability of 0.9999994 that the interval contains $\hat{\eta}$. If $\eta < a$ ($\eta > b$), we conclude that the route is feasible (infeasible). If $\eta \in [a, b]$, no conclusions can be drawn and additional scenarios must be generated to further decrease the size of the $[a, b]$ interval. Algorithm 1 summarizes the procedure.

Algorithm 1 Statistical Inference with Monte Carlo Sampling

Input: set of customers C', a limit on the maximum number of scenario evaluations N
$x \leftarrow 0$ ▷ Number of feasible scenarios
for $n = 1$ to N **do**
  $[d]_{i \in C'} \leftarrow$ randomly generated demand vector
  if $\sum_{i \in C'} d_i \leq Q^{\text{SE}}$ **then**
    $x \leftarrow x + 1$
  $[a, b] \leftarrow$ `Agresti-Coull`(x,n) ▷ Estimate interval $[a, b]$ such that $P\big(\sum_{i \in C'} \xi_i \leq Q^{\text{SE}}\big) \in [a, b]$
  if $\eta < a$ **then return** True
  else if $\eta > b$ **then return** False
end for
return *inconclusive* ▷ Not possible to infer feasibility after N scenario evaluations

Occasionally, it happens that Algorithm 1 terminates with an inconclusive result. Although it is possible to continue the algorithm for larger values of N, we apply the following *two-phase procedure* for checking

feasibility. First, we consider the statistical inference test with Monte Carlo sampling. If this returns inconclusive, the action to be taken depends on the demand distributions of the customers. In the case of independent demand distributions, we convolute the demand distributions as explained in Section 5.1 to compute the exact value of $\hat{\eta}$. Otherwise, we resort to computing the single point estimate (SPE) $\hat{\eta} = x/N$ as our best guess for $\hat{\eta}$, possibly leading to incorrect classification of (in)feasibility. However, if so, the value will be close to η and its impact on the solution quality will be negligible.

5.3 Feasibility Bounds

Regardless of the method, verifying compliance with Constraint (1) is computationally costly. To improve the efficiency of feasibility checks, we introduce the idea of *feasibility bounds*, which enable us to assert compliance with the probabilistic capacity constraint in constant time for most customer combinations. For brevity, we define $\mu(C') = \sum_{i \in C'} \mu_i$ to be the total expected demand of the customers in $C' \subset C$. First, we compute lower and upper bounds, $\underline{\mu}$ and $\bar{\mu}$, expressed in terms of the expected customer demand. For $\underline{\mu}$ it must hold that Constraint (1) is satisfied for all $C' \subset C$ with $\mu(C') < \underline{\mu}$. Similarly, for $\bar{\mu}$ it must hold that Constraint (1) is violated for all $C' \subset C$ with $\mu(C') > \bar{\mu}$. Then, whenever we need to assert whether a set of customers C' in some (partial) route satisfies the chance constraint, we can simply conduct a comparison against the two feasibility bounds as follows. If $\mu(C') < \underline{\mu}$, it is guaranteed that the set of customers C' satisfies Constraint (1). Similarly, if $\mu(C') > \bar{\mu}$, the probability that the total demand of the customers in C' is larger than the vehicle capacity Q^{SE} exceeds η, resulting in a violation of Constraint (1). Finally, only in the event that $\underline{\mu} \leq \mu(C') \leq \bar{\mu}$, compliance with the chance constraint must be verified with the two-phase procedure explained in Section 5.2.

To obtain the value for the lower bound $\underline{\mu}$, starting with $\mu^* = 0$, we search for a customer combination C' with $\mu(C') = \mu^*$ that violates the chance constraint. If no such combination exists, we increase μ^* by one and continue to do so until we reach a value μ^* for which there exists an infeasible customer combination C' with $\mu(C') = \mu^*$. We set the lower bound $\underline{\mu} = \mu^*$. Next, we initialize μ^* with the total mean demand of the customers in the instance, and decrease μ^* until we obtain a value μ^* for which there exists a customer combination C', $\mu(C') = \mu^*$, that satisfies the chance constraint, and set the upper bound $\bar{\mu} = \mu^*$. This procedure is summarized in Algorithm 2.

Algorithm 2 Feasibility Bounds on the Expected Customer Demand

1: **Input:** Set of customers C
2: μ^*, $\underline{\mu}$, $\bar{\mu} = 0$, -1, -1
3: **while** $\underline{\mu} = -1$ **do** ▷ Obtain lower bound $\underline{\mu}$
4: **if** `findInfeasibleCC`(μ^*) **then**
5: $\underline{\mu} = \mu^*$
6: $\mu^* = \mu^* + 1$
7: $\mu^* = \sum_{i \in C} \mu_i$
8: **while** $\bar{\mu} = -1$ **do** ▷ Obtain upper bound $\bar{\mu}$
9: **if** `findFeasibleCC`(μ^*) **then**
10: $\bar{\mu} = \mu^*$
11: $\mu^* = \mu^* - 1$

The problem of identifying (in)feasible customer combinations for a given value of μ^* (Lines 4 and 9 in Algorithm 2) can be formulated as an integer programming problem. Let Ω be a set of scenarios with demand realization ξ_i^ω for customer $i \in C$ in scenario $\omega \in \Omega$. Let x_i be a binary decision variable that

takes the value 1 if customer $i \in C$ is selected. Binary decision variables y_ω are set to 1 if the selected customers violate the vehicle capacity Q^{SE} in scenario $\omega \in \Omega$, and 0 otherwise. The following model is used to determine whether there exists an infeasible customer combination $C^* \subset C$ with $\mu(C^*) = \mu^*$:

$$\begin{aligned}
\text{minimize} \quad & 1 - \frac{1}{|\Omega|} \sum_{\omega \in \Omega} y_\omega, \\
\text{subject to} \quad & \sum_{i \in C} \xi_i^\omega x_i \geq Q^{\text{SE}} - M(1 - y_\omega), \quad \omega \in \Omega, && (6) \\
& \sum_{i \in C} \mu_i x_i = \mu^*, && (7) \\
& x_i \in \{0, 1\}, \qquad i \in C, && (8) \\
& y_\omega \in \{0, 1\}, \qquad \omega \in \Omega. && (9)
\end{aligned}$$

The objective is to minimize the SPE of the feasibility probability over all customer combinations $C^* \subset C$ with $\mu(C^*) = \mu^*$. If the objective value is smaller than η, there exists a combination C^* with $\mu(C^*) = \mu^*$ that violates the chance constraint. Constraints (6) link the x_c and y_ω variables. If the total demand of the selected customers in scenario $\omega \in \Omega$ exceeds the vehicle capacity, y_ω is set to 1. Constraint (7) restricts the combinations of customers C^* to those for which $\mu(C^*) = \mu^*$ holds. Constraints (8) - (9) set the domains of the decision variables.

Similarly, we use the following model to determine whether there exists a customer combination $C^* \subset C$ with $\mu(C^*) = \mu^*$ that satisfies Constraint (1):

$$\begin{aligned}
\text{maximize} \quad & 1 - \frac{1}{|\Omega|} \sum_{\omega \in \Omega} y_\omega, \\
\text{subject to} \quad & \sum_{i \in C} \xi_c^\omega x_i \leq Q^{\text{SE}} + M y_\omega, \quad \omega \in \Omega, \\
& (7) - (9).
\end{aligned}$$

If the objective value is larger than η, there exists a customer combination that satisfies the chance constraint.

When solving the integer programming formulations with a standard solver, it is no longer possible to invoke any of the feasibility tests described in Sections 5.1 and 5.2. Additionally, the size of Ω has a direct impact on the preciseness of the SPE. Therefore, instead of solving this formulation, we proceed with an efficient search technique that enables us to use the feasibility tests.

For a given value of μ^*, we retrieve all customer combinations C^* with $\mu(C^*) = \mu^*$ by solving a perfect subset sum problem with dynamic programming and store the obtained customer combinations in set $\tilde{C}$. Next, we iterate over the customer combinations in $\tilde{C}$ and invoke the two-phase procedure to verify compliance with Constraint (1) (Section 5.2). When searching for a(n) (in)feasible customer combination in $\tilde{C}$, we terminate the search once we identify such a combination and omit evaluating any remaining customer combination in $\tilde{C}$.

Thus far, we discussed bounds expressed in terms of the expected customer demand. With these bounds, we reduce the set of customer combinations for which the probabilistic capacity constraint needs to be verified to customer combinations $C' \subset C$ with $\underline{\mu} \leq \mu(C') \leq \bar{\mu}$. Further reduction of the size of this set is achieved by imposing additional bounds expressed in terms of the total variance of customer demands. For brevity, we define $\sigma^2(C') = \sum_{i \in C'} \sigma_i^2$ to be the total variance of the demands of the customers in $C' \subset C$. For each

value $\mu \in \{\underline{\mu}, \ldots, \bar{\mu}\}$, we derive lower and upper bounds, $\underline{\sigma}^2(\mu)$ and $\bar{\sigma}^2(\mu)$, in a similar way as the bounds on the total expected customer demand. For a given value of μ, we set $\underline{\sigma}^2(\mu)$ to the lowest value of σ^{2*} for which there exists an infeasible customer combination C' with $\mu(C') = \mu$ and $\sigma^2(C') = \sigma^{2*}$. Similarly, $\bar{\sigma}^2(\mu)$ is set to the highest value of σ^{2*} for which there exists a customer combination C', with $\mu(C') = \mu$ and $\sigma^2(C') = \sigma^{2*}$, that satisfies the chance constraint. The procedure is summarized in Appendix 8.1.

The feasibility bounds are computed during a pre-processing phase, prior to the start of the column generation procedure. The value of the upper bound on the total expected customer demand discussed in Section 3 is set to the upper feasibility bound on the mean customer demand, $\bar{Q} = \bar{\mu}$. The feasibility bounds do not require any assumption on the demand distributions of the customers, thus can also be computed on instances with correlated, dependent or data-driven demand distributions.

6 Computational Results

We test the algorithms on instances proposed by Dellaert et al. (2019). These instances represent a circular urban area and are generated for different combinations of depots, satellites and customers. For each combination, we have five instances. Each instance is named "Cbw-x,y,z" where w, x, y, z represent the index, number of depots, number of satellites, and number of customers, respectively. For example, "Cb2-2,3,30" denotes the second instance generated with two depots, three satellites, and 30 customers. The instances with 100 customers are reduced to 75 customers by eliminating the last 25 customers. The expected demand of each customer is equal to the integer demand value in the original instance. We set the FE and SE capacities to $Q^{\text{FE}} = 150$ and $Q^{\text{SE}} = 50$.

In Section 6.1, we present the results on instances with independent demand distributions. Correlated customer demands are considered in Section 6.2. All experiments are performed on a single thread of an Intel® Xeon E5-2696 (2.4 GHz) CPU with 18GB of memory. The (R)MP is solved using IBM® CPLEX® version 12.10. Finally, a time limit of two hours, including pre-processing time, is imposed on each run of the algorithm.

6.1 Independent Demands

We consider under-dispersed, neutral, and over-dispersed uncertain demands by iteratively assigning one of the following demand distributions to each customer: binomial with a variance-to-mean ratio of 0.5; Poisson with a variance-to-mean ratio of 1; or negative binomial with a variance-to-mean ratio of 1.5, 2 or 3.

When verifying the chance constraint, we first consider statistical inference (Section 5.2) with a maximum of $N = 10,000$ scenarios, since, for many customer combinations, performing statistical inference is much quicker than using the convolution procedure outlined in Section 5.1. For relatively few customer combinations the statistical inference test returns inconclusive ($< 0.05\%$ of the cases, on average), and we resort to convoluting the customer demand distributions instead.

The remainder of this section is structured as follows. In Section 6.1.1, we compare the performances of both labeling procedures. In Section 6.1.2, we analyze the impact of the feasibility bounds on the runtime. The characteristics of the solutions are discussed in Section 6.1.3. Finally, in Section 6.1.4, we consider solving different deterministic versions of the problem using the expected customer demands and show that this always results in worse solutions in terms of cost and feasibility of SE routes compared to solving the stochastic formulation.

Table 3: Independent Demands - Summarized Results

	Multi-label				Two-Echelon One-Path			
Setting	#	Gap(%)	Col.	CPU(s)	#	Gap(%)	Col.	CPU(s)
15 customers								
(2,3)	5	3.64	427	3	5	4.50	636	4
(3,5)	5	4.70	682	6	5	6.05	960	16
(6,4)	5	3.40	742	3	5	3.61	1112	8
Overall	15	3.91	617	4	15	4.72	903	9
30 customers								
(2,3)	5	4.73	1584	43	5	4.39	1805	35
(3,5)	5	2.17	1666	55	5	2.69	2236	44
(6,4)	5	3.65	1649	24	5	3.11	2541	51
Overall	15	3.52	1633	41	15	3.39	2194	43
50 customers								
(2,3)	5	2.79	3190	871	5	2.79	3792	339
(3,5)	5	4.19	3194	626	5	3.07	3992	292
(6,4)	5	3.23	3338	205	5	3.2	4702	232
Overall	15	3.40	3241	567	15	3.02	4162	288
75 customers								
(2,3)	0	-	-	-	4	2.30	5433	1778
(3,5)	4	2.92[a]	6322	2889	4	2.66	6204	1741
(6,4)	3	2.54[a]	6765	1396	4	2.31	7432	1970
Overall	7	2.76[a]	6512	2249	12	2.43	6356	1830

[a] Average is taken over the solved instances

6.1.1 Performance comparison of multi-label and 2E-1P algorithms.

In this section, we compare the performance of the two labeling procedures. The chance constraint level is set to $\eta = 0.95$. A summary of the results is presented in Table 3, where columns "#", "Gap", "Col." and "CPU" indicate, respectively, the number of instances that could be solved with the allotted runtime and memory, the average optimality gap, the average number of columns generated, and the average runtime in seconds. Recall that once an optimal solution to RMP is obtained, we compute an upper bound on the MP by solving MP directly with the set of tour-trees generated by the pricing problem. In terms of lower bounds, the bounds computed with the multi-label algorithm are at least as strong as the bounds computed with 2E-1P algorithm, since the former algorithm computes elementary routes, whereas the latter algorithm produces *ng*-routes. When computing optimality gaps, we compare the upper bounds to the best lower bound. Detailed results on a per-instance basis can be found in Appendix 8.2.

Using the multi-label algorithm, we solve the RMP to optimality for 52 out of the 60 instances. The remaining eight instances ran out of memory. Stronger completion bounds are required to solve these instances. With the 2E-1P algorithm, we obtain optimal solutions to the RMP on 57 instances. The remaining two instances ran out of time. The 2E-1P algorithm returns lower bounds that are, on average, 1.2% weaker than the lower bounds returned by the multi-label algorithm.

Figure 5 shows the optimality gaps computed with both labeling procedures. The x and y coordinates of a data-point in the graph equal, respectively, the optimality gaps obtained with the multi-label and 2E-1P algorithms on a unique instance. When both procedures return the same gap, the corresponding data-point is plotted on the diagonal line. If a point falls above (below) the diagonal, a smaller optimality gap is obtained with the multi-label algorithm (2E-1P algorithm). The graph shows that optimality gaps between

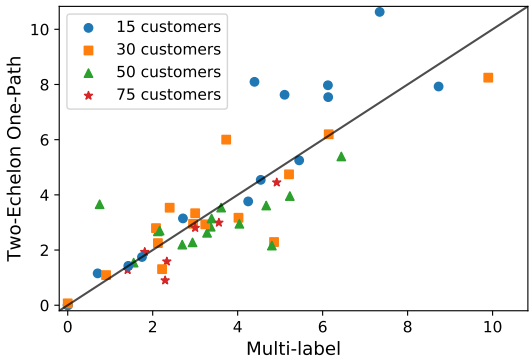

Figure 5: Comparison of optimality gaps

Table 4: Runtime (in seconds) of Multi-label Algorithm with(out) Feasibility Bounds (FB)

$\|C\|$	Without FB	With FB	Speed up factor[b]
15	3.5	3.7	0.89
30	54.1	40.8	1.33
50	938.7	576.2	1.57
75[a]	4287.7	2248.8	1.87

[a] Average taken over the instances on which we obtain optimal solutions to the RMP, [b] Average over the speed up factors per instance.

0 and 11% are obtained. Moreover, none of the algorithms are better for a particular instance size. Finally, regardless of the labeling procedure, the average optimality gap is less than 4%.

On average, the multi-label algorithm requires shorter computation times on instances with 15 and 30 customers. As the number of customers in the instance grows, it becomes more important to efficiently control the growth of the labels at an early stage of the labeling algorithm, which is achieved with dominance rules.

6.1.2 Impact of the feasibility bounds.

We compute feasibility bounds in terms of the total mean and variance of the customer demands to reduce the computation time spent on verifying the probabilistic capacity constraint during the labeling algorithms. In this section, we study the impact of the feasibility bounds by comparing the runtimes required to solve the RMP with the multi-label algorithm and with or without the feasibility bounds. In the latter case, we derive the feasibility bounds to obtain a value for $\bar{Q}$, but do not use the feasibility bounds when verifying the chance constraint. In this way, the same truncated vehicle capacity $\bar{Q}$ is used for the computation of the completion bound in both algorithms and the difference in the runtimes can be attributed to the additional time spent on verifying the probabilistic capacity constraint. Table 4 shows the corresponding runtimes, averaged over the different instance sizes. It can be observed that the impact of the feasibility bounds on the runtime increases with the number of customers in the instance.

Table 5: Structural Characteristics of 2E-VRPSD Solutions

	$\|C\| = 15$	$\|C\| = 30$	$\|C\| = 50$	$\|C\| = 75$
# tour-trees	2.9	4.9	7.5	11.4
# SE routes	7.3	13.5	21.8	32.6
# SE routes in a tour-tree	2.5	2.8	2.9	2.9
# customers in a tour-tree	5.3	6.2	6.7	6.6
# customers in an SE route	2.1	2.2	2.3	2.3

Table 6: Number of Depots and Satellites Used in the Solution

	$\|C\| = 15$			$\|C\| = 30$			$\|C\| = 50$			$\|C\| = 75$		
	(2,3)[a]	(3,5)[a]	(6,4)[a]	(2,3)	(3,5)	(6,4)	(2,3)	(3,5)	(6,4)	(2,3)	(3,5)	(6,4)
$\|D\|^s$	1.8	2.2	2.8	1.8	3.0	3.8	2.0	2.8	3.8	2.0	3.0	3.3
$\|S\|^s$	2.2	2.6	3.0	2.4	4.6	3.8	3.0	5.0	3.8	3.0	5.0	4.0

[a] Number of depots and satellites available, [s] Number of depots and satellites in the solution

6.1.3 Characteristics of 2E-VRPSD solutions.

Table 5 provides an overview of characteristics of the 2E-VRPSD solutions, where the averages are computed for each instance size. We report the average number of tour-trees (FE routes), number of SE routes, number of SE routes in a tour-tree, number of customers in a tour-tree, and number of customers in an SE route. Both the average number of tour-trees and the average number of SE routes increase proportionally to the number of customers in the instance.

The average number of depots and satellites used in the solution are given in Table 6 for different combinations of depots, satellites and customers. The number of satellites used increases with the number of customers in the instances. For the instances with 75 customers, it holds that all satellites are used. The additional satellites that are included in the solutions to the larger instances might require visits from FE vehicles starting from different, additional, depots than those already used in the solutions to the smaller instances, explaining the corresponding increase in the number of depots used.

6.1.4 Value of the stochastic formulation.

As an alternative to solving the 2E-VRPSD, one could solve a simpler, non-stochastic version of the problem using expected customer demands. To try to achieve some level of protection against exceeding the vehicle capacity, some portion of the SE vehicle capacity can be reserved to serve as a safety buffer. More precisely, instead of solving 2E-VRPSD with Constraint (1), we could solve the problem with the following modified capacity constraint:

$$\sum_{i \in C'} \mu_i \leq Q^{\mathrm{SE}} - s, \tag{10}$$

where parameter s, $0 \leq s < Q^{\mathrm{SE}}$, determines the amount of SE vehicle capacity to reserve. We refer to this version of the problem as the truncated expected value problem (TEVP). Clearly, larger values of s lower the probability of exceeding the vehicle capacity, but at the same time increase the number of SE routes in a solution since fewer customers can be serviced in the same route.

In Figure 6, we compare TEVP with our stochastic version of the problem to demonstrate the value of the stochastic formulation. In this experiment, we solve all problem instances with 30 and 50 customers,

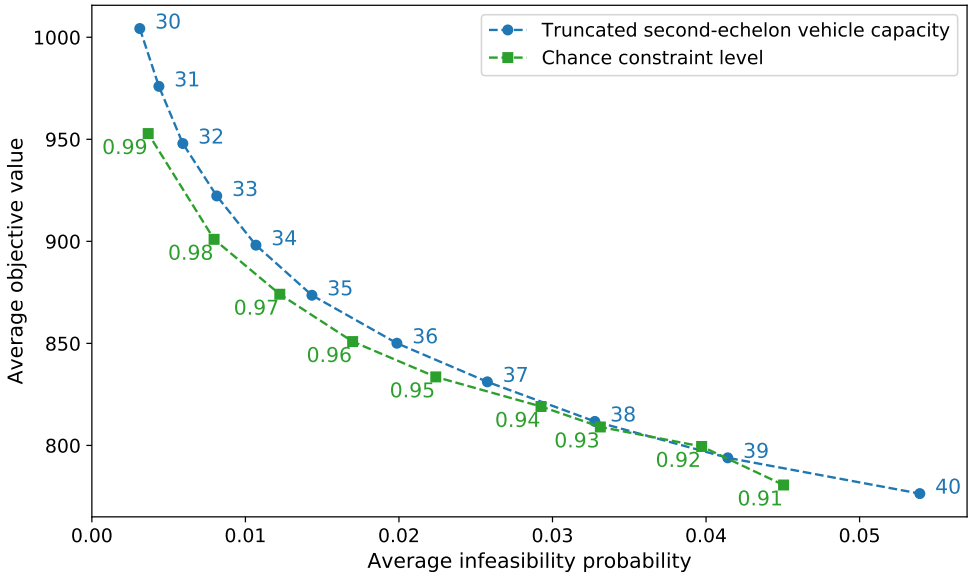

Figure 6: Pareto frontiers showing the value of the stochastic formulation

with both formulations for different values of η and s, using the multi-label algorithm. For each SE route in the solution corresponding to the upper bound, we compute the probability that the total demand of the customers in the SE route exceeds the vehicle capacity (infeasibility probability) through convoluting the customer demand distributions (Section 5.1). For given values of s or η, we plot the average route infeasibility (averaged over all routes in all solutions) against the average solution value.

As can be observed from Figure 6, when we reserve 10 units of capacity ($s = 10$, bottom right blue data-point), the probability of having insufficient vehicle capacity exceeds 5%; when 40% of the vehicle capacity is reserved ($s = 20$, top left blue data-point), the probability of exceeding the vehicle capacity drops to about 0.3%. In contrast, a similar level of protection is achieved with the stochastic formulation for $\eta = 0.99$, but at 4.96% lower cost on average. In fact, it can be observed that the entire chance constraint frontier is below the TEVP frontier for reasonable values of η and s, confirming the superiority of the stochastic formulation over TEVP.

6.2 Correlated Demands

Thus far, only problem instances with independent customer demands are considered. In practice, however, customer demands are often correlated, due to a wide range of external factors that may induce correlations, such as weather, events, and sales (Gendreau et al., 2016). In this section, we consider the same instances as described earlier, and only change the demand distributions for the customers. We investigate the impact of demand correlations on our solution approaches and the 2E-VRPSD solutions. In particular, when demands are positively correlated, we anticipate that the number of SE routes required will increase, since fewer customers can be combined in the same SE route without violating the probabilistic capacity constraint. Conversely, we expect denser SE routes in the solutions on the instances with negatively correlated demands. When including customers in a route that are negatively correlated, the variance of the total demand of the customers in the route decreases. This creates room for cost-effective routes visiting more customers or customers with a higher total expected customer demand than allowed if no correlation was present, without

Table 7: Mixed Correlation

	Group $\mathcal{A}$	Group $\mathcal{B}$
Group $\mathcal{A}$	positive	negative
Group $\mathcal{B}$	negative	positive

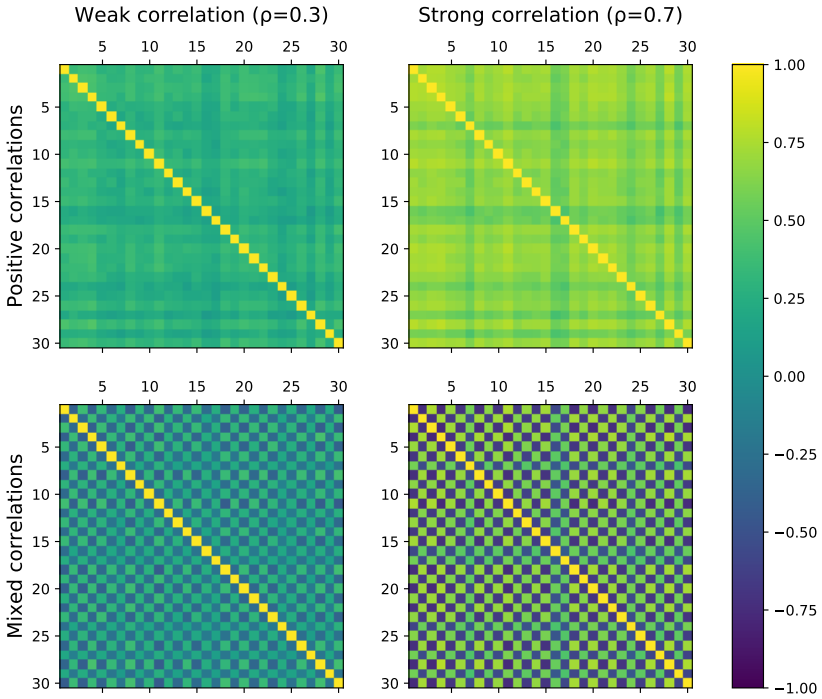

Figure 7: Pearson correlation coefficients between each pair of customers obtained on one instance for the different settings

violating the chance constraint.

We examine two levels of correlation: weak ($\rho = 0.3$) and strong ($\rho = 0.7$) and two types of correlation: positive and mixed. The generation of mixed correlation requires the partition of the customers into two groups. Table 7 shows that mixed correlation implies that the demand of a customer is positively correlated with the demand of customers in the same group, and negatively correlated with the demand of customers in the other group. An example of the different correlations in an instance with 30 customers is given in Figure 7.

The two-phase procedure introduced in Section 5.2 is used to determine whether an SE route meets the probabilistic capacity constraint. Both phases require the generation of demand scenarios where customer demands are randomly sampled from their underlying demand distributions. For the experiments involving customers with correlated demand, instead of explicitly defining correlated demand distributions per customer, we bypass this step and directly generate correlated demand scenarios using the procedure outlined below.

For each customer $i \in C$, we generate Poisson distributed demand scenarios and apply a multiplicative transformation to obtain correlated demand scenarios. Let d_i^ω be the Poisson generated demand realization of customer $i \in C$ in scenario ω. Let α^ω be the correlation parameter in scenario ω, which is uniformly distributed on the interval $[a, b]$, with $[a, b] = [0.7, 1.3]$ if $\rho = 0.3$, and $[a, b] = [0.2, 1.8]$ if $\rho = 0.7$. The

Table 8: Correlated Demand - Summarized Results

Correlated setting		Percentage change relative to the independent case		
		Objective value	Number of SE routes	CPU (s)
Positive	$\rho = 0.3$	9.41	12.51	-28.94
	$\rho = 0.7$	41.82	55.78	-54.83
Mixed	$\rho = 0.3$	2.90	2.21	-8.47
	$\rho = 0.7$	10.78	9.86	16.03

positively correlated demand scenarios are generated as follows:

$$\xi_i^\omega = \alpha^\omega \cdot d_i^\omega. \tag{11}$$

In the case of mixed correlation, we use Equation (11) to generate the demands of the customers in group $\mathcal{A}$ and Equation (12) to generate the demands of the customers in group $\mathcal{B}$:

$$\xi_i^\omega = ((1 - \alpha^\omega) + 1) \cdot d_i^\omega. \tag{12}$$

As stated in Section 4.3, the dominance rules in the 2E-1P algorithm are only valid for independent demand distributions. Therefore, in the remainder of this section, we only consider the multi-label algorithm when solving the instances with correlated demand.

In Table 8, we compare the results obtained with the different correlations to the results obtained on instances with independent Poisson distributed demands (independent case). We report the average increase in objective value (percentage), average increase in the number of SE routes (percentage), and average change in runtime (percentage). The averages are computed over the instances with 30 and 50 customers. Detailed results on a per-instance basis can be found in Appendix 8.3.

On average, higher objective values are observed for all four correlation categories. A similar increase can be witnessed in the number of SE routes needed. Comparing the different correlations, we conclude that about the same increase in the objective value and number of SE routes is obtained with weak positively correlated demands and strong mixed correlated demands. Finally, with mixed correlation, the number of feasible customer combinations increases, leading to more possibilities in the labeling algorithm, which explains the longer runtimes.

As a final experiment, to investigate our earlier expectation that with mixed correlation we have denser SE routes visiting customers from both groups, we study the characteristics of the SE routes in the solutions obtained with the different settings. Next to deriving the average number of customers in an SE route, we compute the average percentage of routes containing customers from both groups (mixed routes) as a percentage of the SE routes visiting more than one customer. These percentages can also be derived on the instances with positive or no correlation under the assumption that the groups exist for those instances as well. Table 9 shows the resulting values for the different settings. As expected, with mixed correlation we have a higher percentage of mixed routes. The average number of customers in an SE route is less with mixed correlation than without any correlation, but higher when compared to the instances with positive correlation.

Table 9: Correlated Demand - SE Route Information

	No correlation	Positive correlation		Mixed correlation	
	$\rho = 0$	$\rho = 0.3$	$\rho = 0.7$	$\rho = 0.3$	$\rho = 0.7$
Average percentage of mixed SE routes	65.38	59.04	48.51	79.68	93.28
Average number of customers in an SE route	2.47	2.22	1.57	2.42	2.25

7 Conclusions

Two-echelon distribution systems are often considered in city logistics to maintain economies of scale and satisfy the cities' emission zone requirements. In this paper, we formulate the 2E-VRPSD as a chance-constrained stochastic optimization problem and consider two solution procedures based on column generation. We propose the multi-label algorithm as a novel labeling algorithm based on simultaneous labeling of SE routes. Additionally, we implement the Two-Echelon One-Path algorithm where SE routes are constructed sequentially. We use statistical inference techniques to speed up feasibility check operations. When these are inconclusive, we consider the convolution of probability distributions in the case of independent demand distributions and resort to single point estimates when demands are correlated. To reduce the computational efforts on validating the chance constraint, we introduce the notion of feasibility bounds and apply them to the sum of the means and variances of the customer demands.

Computational experiments are performed on instances with independent demand distributions and on instances with correlated demand distributions. We compute linear bounds on instances with independent demand distributions and up to 75 customers within a time limit of two hours. Regardless of the labeling procedure, the average optimality gap is less than 4%. We demonstrate that solving the 2E-VRPSD always results in better solutions in terms of objective value and feasibility of SE routes compared to solving the expected value 2E-VRP with some truncated SE vehicle capacity. For example, we achieve a similar protection level at 4.96% lower cost on average if we solve the 2E-VRPSD with a chance constraint level of $\eta = 0.99$ instead of the expected value 2E-VRP with 60% of the original SE vehicle capacity. Finally, we show that with the use of feasibility bounds, the labeling algorithms' runtimes are reduced significantly.

We demonstrate that the multi-label algorithm and feasibility bounds can also be used to solve instances with correlated demand distributions. Next to solving instances with positively correlated demands, we also consider instances with positively and negatively correlated demands. The latter set of instances shows that up to 93% of the SE routes include customers with negatively correlated demands. This is not surprising because, by including customers in a negatively correlated route, the variance of the total demand of the customers in a route decreases, creating room for additional customers or allowing customer combinations with a higher total expected mean compared to the case without correlation.

Future research on the chance-constrained two-echelon vehicle routing problem with stochastic demands is recommended to relax the assumption that an SE vehicle leaves a satellite with full capacity. In this paper, we applied the feasibility bounds to stochastic demands. Further implementation of the feasibility bounds to other uncertain parameters is another promising research direction.

Acknowledgement

This work was funded by The Dutch Research Council (NWO) DAREFUL project under grant 629.002.211 and carried out on the Dutch national e-infrastructure with the support of SURF Cooperative.

8 Appendix

8.1 Feasibility Bounds Expressed in Terms of the Total Variance of the Customer Demands

Algorithm 3 summarizes the procedure of obtaining values for the feasibility bounds expressed in terms of the total variance of customer demands.

Algorithm 3 Feasibility Bounds on the Variance of the Customer Demands

Input: Set $\tilde{C}$ of customer combinations $C' \subset C$ satisfying $\mu(C') = \mu$
σ^{2*}, $\underline{\sigma}^2$, $\bar{\sigma}^2 = 0$, -1, -1
while $\underline{\mu} = -1$ **do** ▷ Obtain lower bound $\underline{\sigma}^2(\mu)$
    if `findInfeasibleCCvar`(μ, σ^{2*}) **then**
        $\underline{\sigma}^2 = \sigma^{2*}$
    $\sigma^{2*} = \sigma^{2*} + 1$
$\sigma^{2*} = \max_{C' \in \tilde{C}} \sum_{i \in C'} \sigma_i^2$
while $\bar{\sigma}^2 = -1$ **do** ▷ Obtain upper bound $\bar{\sigma}^2(\mu)$
    if `findFeasibleCCvar`(μ^*, σ^{2*}) **then**
        $\bar{\sigma}^2 = \sigma^{2*}$
    $\sigma^{2*} = \sigma^{2*} - 1$

8.2 Independent Demand - Detailed Results

Tables 10-13 show the solutions derived on all instances. The number of instances that could be solved with the allotted runtime and memory (#), the lower bound (LB), the upper bound (UB), the average optimality gap (Gap), the average number of columns generated (Col.) and the average running times in seconds (CPU) are given. Recall that once an optimal solution to RMP is obtained, we compute an upper bound on the MP by solving MP directly with the set of tour-trees generated by the pricing problem. In terms of lower bounds, the bounds derived with the multi-label algorithm are, per definition, at least as strong as the bounds derived with 2E-1P algorithm, since the former algorithm computes elementary routes, whereas the latter algorithm produces *ng*-routes. When computing optimality gaps, we compare the upper bounds to the best lower bound, that is, we consider the lower bound returned by the multi-label algorithm if it solved the RMP to optimality, otherwise, we compute the optimality gap with respect to the lower bound returned by the 2E-1P algorithm.

8.3 Correlated Demand - Detailed Results

Tables 14-15 show the solutions obtained when solving the instances with independent Poisson distributed demands, positively correlated demand and mixed correlated demand. For each setting and instance, the lower bound (LB), upper bound (UB), optimality gap in percentages (Gap) and runtime in seconds (CPU) are given.

Table 10: Independent Demand - 15 Customers

	Multi-label					Two-Echelon One-Path				
Instances	LB	UB	Gap(%)	Col.	CPU(s)	LB	UB	Gap(%)	Col.	CPU(s)
Cb1-2,3,15	458.57	486.66	6.12	604	3	455.46	495.13	7.97	621	3
Cb2-2,3,15	313.67	329.68	5.10	322	2	312.79	337.59	7.63	542	3
Cb3-2,3,15	410.87	422.01	2.71	317	2	410.87	423.80	3.15	641	3
Cb4-2,3,15	401.06	418.10	4.25	436	4	391.27	416.16	3.76	679	6
Cb5-2,3,15	339.45	339.45	0.00	455	2	335.91	339.45	0.00	699	5
Average			3.64	427	3			4.50	636	4
Cb1-3,5,15	363.98	380.51	4.54	663	5	345.00	380.51	4.54	922	8
Cb2-3,5,15	426.28	457.58	7.34	651	5	421.25	471.60	10.63	966	6
Cb3-3,5,15	374.98	381.54	1.75	692	4	369.16	381.54	1.75	800	5
Cb4-3,5,15	315.76	329.64	4.40	709	9	296.68	341.32	8.10	1205	53
Cb5-3,5,15	366.69	386.66	5.45	696	4	358.04	385.94	5.25	907	6
Average			4.70	682	6			6.05	960	16
Cb1-6,4,15	360.75	365.89	1.43	788	3	347.28	365.89	1.43	1082	8
Cb2-6,4,15	416.35	416.35	0.00	719	3	403.44	416.35	0.00	969	4
Cb3-6,4,15	384.37	387.08	0.70	731	3	370.02	388.81	1.15	1164	12
Cb4-6,4,15	317.67	337.14	6.13	701	2	301.04	341.62	7.54	1239	6
Cb5-6,4,15	312.54	339.82	8.73	769	4	306.39	337.31	7.93	1108	9
Average			3.40	742	3			3.61	1112	8
Average			3.91	617	4			4.72	903	9

Table 11: Independent Demand - 30 Customers

	Multi-label					Two-Echelon One-Path				
Instances	LB	UB	Gap(%)	Col.	CPU(s)	LB	UB	Gap(%)	Col.	CPU(s)
Cb1-2,3,30	737.16	766.79	4.02	1678	72	736.29	760.53	3.17	1619	30
Cb2-2,3,30	634.18	667.21	5.21	1489	27	633.85	664.26	4.74	1943	40
Cb3-2,3,30	858.76	879.36	2.40	1741	50	851.27	889.11	3.53	1522	33
Cb4-2,3,30	685.81	700.36	2.12	1479	23	685.66	701.24	2.25	1485	19
Cb5-2,3,30	663.38	729.05	9.90	1534	45	643.46	718.10	8.25	2455	52
Average			4.73	1584	43			4.39	1805	35
Cb1-3,5,30	583.63	602.53	3.24	1310	35	575.67	600.74	2.93	2217	28
Cb2-3,5,30	651.18	657.05	0.90	1728	49	638.84	658.29	1.09	2267	42
Cb3-3,5,30	612.31	630.67	3.00	1666	45	593.61	632.72	3.33	2400	42
Cb4-3,5,30	601.53	623.97	3.73	1941	96	599.90	637.64	6.00	2153	80
Cb5-3,5,30	726.72	726.72	0.00	1683	48	715.24	727.21	0.07	2144	30
Average			2.17	1666	55			2.69	2236	44
Cb1-6,4,30	685.40	700.62	2.22	1545	22	671.60	694.36	1.31	2619	44
Cb2-6,4,30	607.26	619.88	2.08	1735	27	594.90	624.23	2.79	2492	85
Cb3-6,4,30	613.71	643.52	4.86	1625	19	611.41	627.77	2.29	2438	33
Cb4-6,4,30	556.80	573.23	2.95	1565	24	547.47	573.23	2.95	2886	60
Cb5-6,4,30	592.02	628.38	6.14	1774	29	588.92	628.70	6.20	2270	33
Average			3.65	1649	24			3.11	2541	51
Average			3.52	1633	41			3.39	2194	43

Table 12: Independent Demand - 50 Customers

	Multi-label					Two-Echelon One-Path				
Instances	LB	UB	Gap(%)	Col.	CPU(s)	LB	UB	Gap(%)	Col.	CPU(s)
Cb1-2,3,50	1118.91	1142.66	2.12	2704	571	1116.37	1148.87	2.68	3915	169
Cb2-2,3,50	1162.73	1204.72	3.61	3457	2365	1162.73	1203.88	3.54	4346	677
Cb3-2,3,50	1181.54	1213.39	2.70	3493	472	1180.81	1207.49	2.20	3468	257
Cb4-2,3,50	1161.60	1186.72	2.16	3406	561	1160.20	1193.02	2.71	3651	380
Cb5-2,3,50	1045.38	1080.56	3.37	2892	386	1043.71	1075.12	2.85	3580	211
Average			2.79	3190	871			2.79	3792	339
Cb1-3,5,50	943.36	992.68	5.23	3510	1743	938.11	980.70	3.96	4913	485
Cb2-3,5,50	989.95	1019.03	2.94	2892	270	987.24	1012.51	2.28	3927	240
Cb3-3,5,50	1043.17	1059.35	1.55	3483	362	1034.70	1059.27	1.54	3421	104
Cb4-3,5,50	915.57	959.56	4.80	3266	329	908.40	935.37	2.16	3924	142
Cb5-3,5,50	855.88	911.00	6.44	2820	427	848.93	902.03	5.39	3773	492
Average			3.49	3192	749			2.93	3892	315
Cb1-6,4,50	930.40	968.00	4.04	3388	242	930.21	957.87	2.95	5333	236
Cb2-6,4,50	1064.19	1099.09	3.28	3372	208	1059.67	1092.13	2.63	3939	152
Cb3-6,4,50	963.06	970.29	0.75	3940	184	959.74	998.31	3.66	4983	235
Cb4-6,4,50	873.25	902.81	3.39	2872	149	868.74	900.70	3.14	4808	228
Cb5-6,4,50	944.09	988.20	4.67	3116	241	943.88	978.21	3.61	4445	308
Average			3.23	3338	205			3.20	4702	232
Average			3.40	3241	567			3.02	4162	288

Table 13: Independent Demand - 75 Customers

	Multi-label					Two-Echelon One-Path				
Instances	LB	UB	Gap(%)	Col.	CPU(s)	LB	UB	Gap(%)	Col.	CPU(s)
Cb1-2,3,75	-	-	-	-	-	-	1453.36	-	6636	7200
Cb2-2,3,75	-	-	-	-	-	1617.50	1656.35	2.40	5041	2077
Cb3-2,3,75	-	-	-	-	-	1620.37	1644.94	1.52	5929	1205
Cb4-2,3,75	-	-	-	-	-	1827.09	1903.60	4.19	5004	2813
Cb5-2,3,75	-	-	-	-	-	1649.08	1667.11	1.09	5759	1019
Average			-	-	-			2.30	5433	1778
Cb1-3,5,75	1499.90	1573.69	4.92	6043	3372	1493.86	1566.65	4.45	6168	2132
Cb2-3,5,75	1420.05	1470.57	3.56	5678	3521	1418.37	1462.55	2.99	6199	1552
Cb3-3,5,75	-	-	-	-	-	-	1471.92	-	6930	7200
Cb4-3,5,75	1639.09	1662.21	1.41	6422	2396	1634.75	1660.02	1.28	6033	1333
Cb5-3,5,75	1687.13	1717.71	1.81	7147	2267	1687.13	1719.74	1.93	6416	1948
Average			2.92	6322	2889			2.66	6204	1741
Cb1-6,4,75	-	-	-	-	-	-	-	-	-	-
Cb2-6,4,75	-	-	-	-	-	1445.72	1503.08	3.97	7278	2315
Cb3-6,4,75	1545.06	1580.49	2.29	6106	1474	1544.13	1558.97	0.90	7509	2591
Cb4-6,4,75	1564.76	1611.71	3.00	6855	1702	1562.19	1608.66	2.81	6729	722
Cb5-6,4,75	1485.96	1520.62	2.33	7335	1011	1485.36	1509.52	1.59	8210	2252
Average			2.54	6765	1396			2.31	7432	1970
Average			2.76	6512	2249			2.43	6356	1830

Table 14: Correlated Demand - Mixed Correlation

	No correlation ($\rho = 0$)				Positive correlation $\rho = 0.3$				Positive correlation $\rho = 0.7$			
Instance	LB	UB	Gap[a]	CPU[b]	LB	UB	Gap	CPU	LB	UB	Gap	CPU
Cb1-2,3,30	686.30	709.98	3.45	23	760.81	768.67	1.03	18	983.01	1006.05	2.34	9
Cb2-2,3,30	619.93	654.05	5.50	28	680.92	703.71	3.35	19	883.49	908.30	2.81	11
Cb3-2,3,30	843.54	862.69	2.27	21	927.91	945.29	1.87	17	1161.97	1171.69	0.84	11
Cb4-2,3,30	654.55	676.43	3.34	21	743.32	755.81	1.68	19	955.21	970.50	1.60	10
Cb5-2,3,30	648.55	655.18	1.02	50	687.89	735.84	6.97	19	873.81	911.14	4.27	12
Cb1-3,5,30	567.63	581.80	2.50	37	613.66	645.22	5.14	36	775.17	807.33	4.15	23
Cb2-3,5,30	627.79	631.89	0.65	47	662.04	662.04	0	43	791.59	816.38	3.13	24
Cb3-3,5,30	579.89	610.23	5.23	45	647.30	653.57	0.97	34	811.81	817.23	0.67	25
Cb4-3,5,30	594.99	627.39	5.45	54	624.35	652.06	4.44	52	806.92	806.92	0	27
Cb5-3,5,30	691.76	694.09	0.34	68	756.53	780.70	3.19	38	971.91	1003.24	3.22	25
Cb1-6,4,30	653.97	662.70	1.34	28	684.03	692.56	1.25	24	881.26	897.00	1.79	13
Cb2-6,4,30	603.46	622.74	3.19	42	620.05	624.13	0.66	23	771.82	808.16	4.71	15
Cb3-6,4,30	582.78	605.82	3.95	27	641.03	647.21	0.96	21	844.11	858.36	1.69	13
Cb4-6,4,30	544.89	557.59	2.33	27	590.02	606.72	2.83	21	740.14	763.79	3.20	13
Cb5-6,4,30	569.39	595.87	4.65	25	619.76	653.53	5.45	31	844.00	848.04	0.48	15
Cb1-2,3,50	1064.64	1118.78	5.09	165	1169.62	1219.78	4.29	90	1559.06	1593.02	2.18	63
Cb2-2,3,50	1099.00	1149.13	4.56	389	1238.41	1295.36	4.60	151	1642.23	1642.23	0	73
Cb3-2,3,50	1087.00	1121.61	3.18	164	1243.58	1273.34	2.39	82	1629.57	1654.47	1.53	59
Cb4-2,3,50	1144.37	1175.38	2.71	206	1236.27	1324.67	7.15	102	1640.62	1655.31	0.90	69
Cb5-2,3,50	989.78	1041.52	5.23	158	1095.37	1132.82	3.42	96	1434.36	1472.34	2.65	65
Cb1-3,5,50	897.94	951.07	5.92	504	1008.64	1029.54	2.07	245	1299.50	1328.75	2.25	180
Cb2-3,5,50	939.12	949.13	1.07	375	1032.29	1063.95	3.07	210	1336.36	1361.81	1.90	152
Cb3-3,5,50	981.64	1036.87	5.63	260	1099.03	1144.10	4.10	216	1481.27	1503.96	1.53	160
Cb4-3,5,50	888.85	922.68	3.81	315	972.34	1016.06	4.50	208	1237.03	1248.66	0.94	147
Cb5-3,5,50	812.21	834.80	2.78	315	896.74	940.72	4.90	287	1175.49	1204.06	2.43	175
Cb1-6,4,50	904.02	937.96	3.75	213	992.68	1029.86	3.75	117	1289.56	1293.56	0.31	84
Cb2-6,4,50	1019.43	1057.57	3.74	119	1123.27	1141.25	1.60	106	1498.71	1499.76	0.07	81
Cb3-6,4,50	928.70	964.47	3.85	202	1023.93	1058.17	3.34	127	1317.85	1340.76	1.74	91
Cb4-6,4,50	812.65	866.18	6.59	186	883.97	948.25	7.27	119	1212.27	1230.52	1.51	84
Cb5-6,4,50	896.63	939.03	4.73	157	980.65	1007.56	2.74	115	1285.30	1310.58	1.97	89

[a] Gap is given in percentages, [b] Runtime is given in seconds.

Table 15: Correlated Demand - No Correlation and Positive Correlation

	Mixed correlation ($\rho = 0.3$)				Mixed correlation $\rho = 0.7$			
Instance	LB	UB	Gap[a]	CPU[b]	LB	UB	Gap	CPU
Cb1-2,3,30	706.90	732.85	3.67	20	770.51	798.66	3.65	50
Cb2-2,3,30	627.36	656.51	4.65	20	684.53	705.47	3.06	32
Cb3-2,3,30	870.61	884.51	1.60	18	966.32	966.32	0	20
Cb4-2,3,30	700.07	716.02	2.28	24	767.57	773.63	0.79	19
Cb5-2,3,30	660.65	677.74	2.59	35	701.25	729.54	4.03	35
Cb1-3,5,30	589.63	602.04	2.10	45	633.40	645.42	1.90	44
Cb2-3,5,30	644.12	644.44	0.05	45	674.96	674.96	0	44
Cb3-3,5,30	603.66	631.99	4.69	42	634.78	638.98	0.66	39
Cb4-3,5,30	603.26	632.26	4.81	47	642.41	650.58	1.27	100
Cb5-3,5,30	720.07	720.57	0.07	53	760.82	767.24	0.84	50
Cb1-6,4,30	654.52	662.76	1.26	24	695.69	695.69	0	27
Cb2-6,4,30	605.17	613.23	1.33	31	638.62	655.97	2.72	36
Cb3-6,4,30	618.62	642.73	3.90	25	671.60	683.54	1.78	26
Cb4-6,4,30	546.43	559.44	2.38	22	578.14	594.58	2.84	21
Cb5-6,4,30	571.01	614.50	7.62	24	634.05	657.57	3.71	27
Cb1-2,3,50	1113.29	1144.70	2.82	195	1182.26	1248.56	5.61	417
Cb2-2,3,50	1134.10	1217.06	7.31	351	1224.23	1257.71	2.73	545
Cb3-2,3,50	1124.77	1177.04	4.65	160	1261.79	1279.62	1.41	156
Cb4-2,3,50	1170.18	1199.72	2.52	164	1253.61	1269.41	1.26	267
Cb5-2,3,50	1015.95	1043.14	2.68	122	1078.74	1138.77	5.56	145
Cb1-3,5,50	938.01	981.83	4.67	450	987.03	1041.18	5.49	614
Cb2-3,5,50	983.20	1025.32	4.28	254	1061.51	1095.73	3.22	288
Cb3-3,5,50	1008.79	1047.33	3.82	269	1104.54	1142.00	3.39	344
Cb4-3,5,50	901.76	943.97	4.68	264	967.24	1000.31	3.42	331
Cb5-3,5,50	840.80	887.74	5.58	277	907.18	942.22	3.86	406
Cb1-6,4,50	920.66	932.37	1.27	190	1005.25	1016.42	1.11	238
Cb2-6,4,50	1057.55	1093.42	3.39	146	1144.70	1172.01	2.39	180
Cb3-6,4,50	950.49	988.88	4.04	227	1017.35	1050.01	3.21	190
Cb4-6,4,50	844.40	890.83	5.50	201	903.09	949.74	5.17	193
Cb5-6,4,50	922.57	930.48	0.86	132	1005.89	1018.61	1.26	208

[a] Gap is given in percentages, [b] Runtime is given in seconds.

References

Agatz, N.A., Fleischmann, M., Van Nunen, J.A., 2008. E-fulfillment and multi-channel distribution–a review. European Journal of Operational Research 187, 339–356.

Anderluh, A., Hemmelmayr, V.C., Nolz, P.C., 2017. Synchronizing vans and cargo bikes in a city distribution network. Central European Journal of Operations Research 25, 345–376.

Anderluh, A., Nolz, P.C., Hemmelmayr, V.C., Crainic, T.G., 2021. Multi-objective optimization of a two-echelon vehicle routing problem with vehicle synchronization and 'grey zone' customers arising in urban logistics. European Journal of Operational Research 289, 940–958.

Baldacci, R., Mingozzi, A., Roberti, R., 2011. New route relaxation and pricing strategies for the vehicle routing problem. Operations Research 59, 1269–1283.

Baldacci, R., Mingozzi, A., Roberti, R., Calvo, R.W., 2013. An exact algorithm for the two-echelon capacitated vehicle routing problem. Operations Research 61, 298–314.

Belgin, O., Karaoglan, I., Altiparmak, F., 2018. Two-echelon vehicle routing problem with simultaneous pickup and delivery: Mathematical model and heuristic approach. Computers & Industrial Engineering 115, 1–16.

Breunig, U., Baldacci, R., Hartl, R.F., Vidal, T., 2019. The electric two-echelon vehicle routing problem. Computers & Operations Research 103, 198–210.

Breunig, U., Schmid, V., Hartl, R.F., Vidal, T., 2016. A large neighbourhood based heuristic for two-echelon routing problems. Computers & Operations Research 76, 208–225.

Brown, L.D., Cai, T.T., DasGupta, A., 2001. Interval estimation for a binomial proportion. Statistical Science , 101–117.

Costa, L., Contardo, C., Desaulniers, G., 2019. Exact branch-price-and-cut algorithms for vehicle routing. Transportation Science 53, 946–985.

Crainic, T.G., Ricciardi, N., Storchi, G., 2004. Advanced freight transportation systems for congested urban areas. Transportation Research Part C: Emerging Technologies 12, 119–137.

Cuda, R., Guastaroba, G., Speranza, M.G., 2015. A survey on two-echelon routing problems. Computers & Operations Research 55, 185–199.

Dellaert, N., Dashty Saridarq, F., Van Woensel, T., Crainic, T.G., 2019. Branch-and-price–based algorithms for the two-echelon vehicle routing problem with time windows. Transportation Science 53, 463–479.

Dellaert, N., Van Woensel, T., Crainic, T.G., Saridarq, F.D., 2021. A multi-commodity two-echelon capacitated vehicle routing problem with time windows: Model formulations and solution approach. Computers & Operations Research 127, 105154.

Dinh, T., Fukasawa, R., Luedtke, J., 2018. Exact algorithms for the chance-constrained vehicle routing problem. Mathematical Programming 172, 105–138.

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

Enthoven, D.L., Jargalsaikhan, B., Roodbergen, K.J., uit het Broek, M.A., Schrotenboer, A.H., 2020. The two-echelon vehicle routing problem with covering options: City logistics with cargo bikes and parcel lockers. Computers & Operations Research 118, 104919.

Feillet, D., Dejax, P., Gendreau, M., Gueguen, C., 2004. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. Networks: An International Journal 44, 216–229.

Florio, A.M., Hartl, R.F., Minner, S., 2020. New exact algorithm for the vehicle routing problem with stochastic demands. Transportation Science 54, 1073–1090.

Florio, A.M., Hartl, R.F., Minner, S., Salazar-González, J.J., 2021. A branch-and-price algorithm for the vehicle routing problem with stochastic demands and probabilistic duration constraints. Transportation Science 55, 122–138.

Gendreau, M., Jabali, O., Rei, W., 2016. 50th anniversary invited article—future research directions in stochastic vehicle routing. Transportation Science 50, 1163–1173.

Grangier, P., Gendreau, M., Lehuédé, F., Rousseau, L.M., 2016. An adaptive large neighborhood search for the two-echelon multiple-trip vehicle routing problem with satellite synchronization. European Journal of Operational Research 254, 80–91.

Jacobsen, S.K., Madsen, O.B., 1980. A comparative study of heuristics for a two-level routing-location problem. European Journal of Operational Research 5, 378–387.

Jie, W., Yang, J., Zhang, M., Huang, Y., 2019. The two-echelon capacitated electric vehicle routing problem with battery swapping stations: Formulation and efficient methodology. European Journal of Operational Research 272, 879–904.

Kitjacharoenchai, P., Min, B.C., Lee, S., 2020. Two echelon vehicle routing problem with drones in last mile delivery. International Journal of Production Economics 225, 107598.

Li, H., Wang, H., Chen, J., Bai, M., 2020. Two-echelon vehicle routing problem with time windows and mobile satellites. Transportation Research Part B: Methodological 138, 179–201.

Liu, R., Tao, Y., Hu, Q., Xie, X., 2017. Simulation-based optimisation approach for the stochastic two-echelon logistics problem. International Journal of Production Research 55, 187–201.

Louveaux, F.V., Salazar-González, J.J., 2018. Exact approach for the vehicle routing problem with stochastic demands and preventive returns. Transportation Science 52, 1463–1478.

Lurkin, V., Hambuckers, J., Van Woensel, T., 2021. Urban low emissions zones: A behavioral operations management perspective. Transportation Research Part A: Policy and Practice 144, 222–240.

Marques, G., Sadykov, R., Deschamps, J.C., Dupas, R., 2020. An improved branch-cut-and-price algorithm for the two-echelon capacitated vehicle routing problem. Computers & Operations Research 114, 104833.

Mühlbauer, F., Fontaine, P., 2021. A parallelised large neighbourhood search heuristic for the asymmetric two-echelon vehicle routing problem with swap containers for cargo-bicycles. European Journal of Operational Research 289, 742–757.

Noorizadegan, M., Chen, B., 2018. Vehicle routing with probabilistic capacity constraints. European Journal of Operational Research 270, 544–555.

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.

Oyola, J., Arntzen, H., Woodruff, D.L., 2018. The stochastic vehicle routing problem, a literature review, part I: models. EURO Journal on Transportation and Logistics 7, 193–221.

Perboli, G., Tadei, R., Vigo, D., 2011. The two-echelon capacitated vehicle routing problem: models and math-based heuristics. Transportation Science 45, 364–380.

Salavati-Khoshghalb, M., Gendreau, M., Jabali, O., Rei, W., 2019a. An exact algorithm to solve the vehicle routing problem with stochastic demands under an optimal restocking policy. European Journal of Operational Research 273, 175–189.

Salavati-Khoshghalb, M., Gendreau, M., Jabali, O., Rei, W., 2019b. A rule-based recourse for the vehicle routing problem with stochastic demands. Transportation Science 53, 1334–1353.

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

Statista, 2020. Global retail e-commerce market size 2014-2023. URL: https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/.

TLN, 2021. Nóg meer steden maken zero-emissie zone bekend. URL: https://www.tln.nl/nieuws/steeds-meer-steden-maken-zero-emissie-zone-bekend/.

Transport Environment, 2018. City bans are spreading in europe. URL: https://www.transportenvironment.org/sites/te/files/publications/City%20bans%20are%20spreading%20in%20Europe_Report.PDF.

Wang, K., Lan, S., Zhao, Y., 2017a. A genetic-algorithm-based approach to the two-echelon capacitated vehicle routing problem with stochastic demands in logistics service. Journal of the Operational Research Society 68, 1409–1421.

Wang, K., Shao, Y., Zhou, W., 2017b. Matheuristic for a two-echelon capacitated vehicle routing problem with environmental considerations in city logistics service. Transportation Research Part D: Transport and Environment 57, 262–276.

Zeng, Z.Y., Xu, W.S., Xu, Z.Y., Shao, W.H., 2014. A hybrid GRASP+VND heuristic for the two-echelon vehicle routing problem arising in city logistics. Mathematical Problems in Engineering 2014.

Zhou, L., Baldacci, R., Vigo, D., Wang, X., 2018. A multi-depot two-echelon vehicle routing problem with delivery options arising in the last mile distribution. European Journal of Operational Research 265, 765–778.