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

Citation for published version (APA):

Document license:
TAVERNE

DOI:
10.1287/trsc.2022.1162

Document status and date:
Published: 01/01/2023

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.
A Chance-Constrained Two-Echelon Vehicle Routing Problem with Stochastic Demands

Natasja Sluijk, Alexandre M. Florio, Joris Kinable, Nico Dellaert, Tom Van Woensel

1. Introduction

The retail e-commerce sales worldwide amounted to 3.53 trillion US dollars in 2019 and are projected to grow to 6.39 trillion US dollars in 2024 (Statista 2021). At the same time, the last mile is becoming increasingly congested, and sustainability is receiving more attention (Agatz, Fleischmann, and Van Nunen 2008). An increasing number of cities in the European Union 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 (Transport Environment 2018, Lurkin, Hambuckers, and Van Woensel 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, for example, with low emission vehicles or electric vehicles with limited range.

The deterministic 2E-VRP is a well-studied problem (Cuda, Guastaroba, and Speranza 2015, Sluijk et al. 2022). In reality, however, various parameters of the problem may be uncertain, for example, travel times, customer demands, and/or customer presence. Although many papers consider stochastic variants of the classical single-echelon vehicle routing problem (Oyola, Arntzen, and Woodruff 2018), only two papers deal with stochastic variants of the 2E-VRP. Both Liu et al. (2017) and Wang, Lan, and Zhao (2017) 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 recourse 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 the 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 satellites and/or depots in a two-echelon network.

In summary, this paper brings the following contributions:

- We formulate the 2E-VRPSD as a chance-constrained stochastic optimization problem.
- We 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 tests to verify that the probabilistic capacity constraint is satisfied. Additionally, we impose feasibility bounds on the stochastic customer demands to reduce the number of customer combinations for which the chance constraint needs to be verified. Both procedures can also be applied to dependent, correlated or data-driven (scenario-based) demand distributions. Our experiments show that with the feasibility 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, Arntzen, and Woodruff 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, Laporte, and Trudeau 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 (Louveaux and Salazar-González 2018, Salavati-Khoshghalb et al. 2019a, Florio, Hartl, and Minner 2020). Finally, the reoptimization policy reoptimizes the route after the current customer demand is revealed, including possible replenishment trips (Secomandi and Margot 2009). In addition, 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.

Stochasticity in customer demand can also be captured with chance constraints. Chance constraints ensure that the total demand of the customers visited in a route fits within the vehicle capacity with high probability. 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, Fukasawa, and Luebbecke (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. Valid inequalities based on quantiles of the customer demand distributions are proposed and the model is solved with branch-and-cut and branch-cut-and-price algorithms.

An overview of solution methods for stochastic vehicle routing problems can be found in Oyola, Arntzen, and Woodruff (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, Jabali, and Rei 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 three decades later, a formal description of the problem appeared in Crainic, Ricciardi, and Storchi (2009). 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, Karaoglan, and Altiparmak 2018); multiple commodities (Dellaert et al. 2021); gray zone customers (Anderluh et al. 2021); and satellite synchronization and multiple trips (Grangier et al. 2016). An extensive overview of various 2E-VRP applications and common solution approaches is provided in Cuda, Guastaroba, and Speranza (2015) and Sluijk et al. (2022).

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, Tadei, and Vigo 2011), large neighborhood search (Wang, Shao, and Zhou 2017, Kitjacharoenchai, Min, and Lee 2020), adaptive large neighborhood search (Breunig et al. 2016, Grangier et al. 2016, Jie et al. 2019, Enthoven et al. 2020, Li et al. 2020), variable neighborhood descent (Belgin, Karaoglan, and Altiparmak 2018), greedy randomized adaptive search procedure (Zeng et al. 2014, Anderluh, Hemmelmayr, and Nolz 2017), and population-based heuristics (Zhou et al. 2018).

Although 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, Lan, and Zhao (2017) 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 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, Lan, and Zhao (2017) and Liu et al. (2017), we capture the stochasticity with chance constraints and do not place strong assumptions 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 $E^{FE} = \{(i, j) \mid i \in D \cup S, j \in S, i \neq j\}$ consists of the edges that can be traversed by first-echelon (FE) vehicles. Similarly, set $E^{SE} = \{(i, j) \mid i \in S \cup C, j \in C, i \neq j\}$ contains the connections that can be used by 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 a sufficiently large homogeneous fleet of vehicles, with capacities $Q^{FE}$ and $Q^{SE}$, respectively, where $Q^{FE} > Q^{SE}$.

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. We assume that each SE vehicle starts its route with full capacity. 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. 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 is at most equal to the vehicle capacity \( Q^{SE} \) with a probability of least \( \eta \), where \( \eta \) is a user-defined parameter to the problem. The demand \( \xi_i \) of each customer \( i \in C \) is a nonnegative integer random variable with mean \( \mu_i \), variance \( \sigma_i^2 \), and \( \mathbb{P}(\xi_i \leq Q^{SE}) \geq \eta \). We do not place any further restrictions on the demand distributions. In particular, the demand of different customers can be statistically dependent. 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 \( C' = \{n_1, \ldots, n_m\} \).

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 \( m \geq 1 \), \( n_0 = n_{m+1} \in D \), \( n_k \in S \) and \( R_k \) representing the set of SE routes starting at satellite \( n_k \), \( k \in \{1, \ldots, m\} \).

The objective of the 2E-VRPSD is to find a set of tour-trees of minimum total 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))), (ii), and (iv,(d,(d,3,4,d),(d,2,c),i)).

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 one 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:

\[
\begin{align*}
\text{(MP)} & \quad \text{minimize} & & \sum_{t \in T} c_t x_t, \\
& \quad \text{subject to} & & \sum_{t \in T} a_{it} x_t = 1, \quad i \in C, \\
& & & x_t \in \{0, 1\}, \quad t \in T. \quad (2)
\end{align*}
\]

The objective function minimizes the total cost of the selected tour-trees. Constraints (2) specify that each customer \( i \in C \) must be 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. As such, problem MP 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. The advantage of this column generation approach, as will become apparent in Section 4 of this paper, is that the complexity related to the stochastic customer demands can be handled entirely inside the
algorithm used to generate new columns, thereby avoiding the need for a stochastic MIP formulation.

To strengthen the formulation, we include a valid inequality that imposes a lower bound on the number of tour-trees that must be selected in an optimal solution. This inequality takes the general form of

\[ \sum_{i \in T} x_i \geq \gamma. \]  

(3)

A detailed description on how to calculate \( \gamma \) is given in Section 5.3.

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

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. To formally define the pricing problem, let \( \lambda_i \), \( i \in C \), and \( \delta \) be the dual values corresponding to Constraints (2) and (3), respectively. The reduced cost \( c'_i \) of a tour-tree \( t \) is given by

\[ c'_i = c_i - \sum_{i \in C} a_{ij} \lambda_i - \delta. \]

The objective of the pricing problem is to find a tour-tree \( t \) with negative reduced cost, that is, with \( c'_i < 0 \). An arc-based formulation of the pricing problem is included in Online Appendix A.

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 nonpromising paths. Two techniques that are commonly 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, Contardo, and Desaulniers 2019). Labels with positive completion bounds are eliminated since they represent partial paths that can never be extended to complete paths with negative reduced cost. 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 for efficiently solving 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. Multilabel Algorithm

In this section, we propose a novel labeling algorithm for solving the pricing problem. Each label in the algorithm represents a partial tour-tree and has both general and SE route specific attributes, hence the name multilabel. Because the information related to the partial SE paths is stored in separate attributes, it is possible to extend different paths simultaneously. In contrast, existing labeling algorithms for constructing a tour-tree for the 2E-VRP consider SE routes consecutively, that is, a new SE route can only be started after the previous SE route has been completed. The main benefit of the multilabel procedure is that the number of SE routes is fixed beforehand. For that reason, we can compute the completion bound on a tour-tree by computing the completion bounds on the individual SE routes and add them together. This enables us to implement the multilabel algorithm without dominance rules, making it fit for dependent, correlated, and data-driven demand distributions.

The multilabel algorithm starts with generating all possible permutations of the satellites that could be visited in an FE route, where each satellite represents the start of a different SE route. Recall that we assume that each SE vehicle starts its route with full capacity. Therefore, the maximum number of SE vehicles that can be supplied by one FE vehicle is equal to \( \theta = |Q^F| = |Q^{SE}| \) and, as such, each permutation consists of at most \( \theta \) satellites. Moreover, following Baldacci et al. (2013), we assume that the number of satellites \( |S| \) is limited. Hence, for reasonable values of \( |S| \) and \( \theta \), we can enumerate all possible \( |S|^\theta \) permutations. We refer to a permutation as a configuration:

**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 repetition of satellites. For example, the configurations of the tour-trees depicted in Figure 1 are \{b, d\}, \{c, e\}, and \{d, c\}.

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

In each iteration of the multilabel algorithm, we randomly select the next multilabel \(\mathcal{L}\) to be extended. For each path \(\mathcal{L}_k, k \in \{1, \ldots, |\mathcal{L}(\Pi)|\}\), we iteratively select a customer \(i \in C\) and check whether the extension of path \(\mathcal{L}_k\) 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 \(\mathcal{L}_k\) is extended to customer \(i\), resulting in a new multilabel \(\mathcal{L}'\). To obtain this multilabel, we copy all attributes of multilabel \(\mathcal{L}\), and update the general attributes and the attributes related to path \(\mathcal{L}'_k\) afterward:

\[
\mathcal{L}' = \mathcal{L},
\]

\[
\mathcal{L}'(\phi) = \mathcal{L}(\phi) + c_{\mathcal{L}'(n_i), i} - \lambda_i,
\]

\[
\mathcal{L}'(\Psi) = \mathcal{L}(\Psi) \cup \{i\},
\]

\[
\mathcal{L}'(n_k) = i,
\]

\[
\mathcal{L}'(\bar{\phi}_k) = \mathcal{L}(\bar{\phi}_k) + c_{\mathcal{L}'(n_i), i} - \lambda_i,
\]

\[
\mathcal{L}'(\bar{\mu}_k) = \mathcal{L}(\bar{\mu}_k) + \mu_i,
\]

\[
\mathcal{L}'(\bar{\sigma}^2_k) = \mathcal{L}(\bar{\sigma}^2_k) + \sigma_i^2,
\]

\[
\mathcal{L}'(m_k) = \mathcal{L}(m_k) + 1.
\]

After extending multilabel \(\mathcal{L}\) to \(\mathcal{L}'\), we compute the reduced cost of the corresponding tour-tree \(t\) by including the routing cost of returning each path to its starting satellite:

\[
c'_i = c_{\mathcal{L}'}(\phi) + \sum_{k=1}^{|\mathcal{L}(\Pi)|} c_{\mathcal{L}'(n_k), n_k} - \delta. \tag{4}
\]

If the total reduced cost is negative, we obtain the complete tour-tree from the multilabel by adding the corresponding satellites to the end of each path and add the tour-tree to the RMP.

### 4.2.1. Symmetry Breaking Rules

The multilabel algorithm, when left untreated, is likely to produce a large number of symmetrical tour-trees, for example, a pair of tour-trees with mirrored SE routes. To reduce the occurrences of symmetry in the multilabel 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 \(A(\mathcal{L}_k)\) and \(C(\mathcal{L}_k)\) to be the first and last customer visited in path \(\mathcal{L}_k\), respectively, for \(k \in \{1, \ldots, |\mathcal{L}(\Pi)|\}\). To express the various symmetry breaking rules, we impose some arbitrary ordering on the elements of \(C\), for example, by assigning each customer a unique number. We write \(i < j\) to denote that customer \(i\) precedes customer \(j\) in this ordering.

The first example is given in Figure 2, where two multilabels, with paths starting from the same satellite, are given. In the first iteration of the multilabel algorithm, customer 1 is added to \(\mathcal{L}_1\) and \(\mathcal{L}'_1\). In the second iteration, customer 2 is added to \(\mathcal{L}_2\) and \(\mathcal{L}'_1\). This results in two multilabels 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.

**Rule 1.** If \(A(\mathcal{L}_k) = A(\mathcal{L}_{k+1})\), then \(A(\mathcal{L}_k) < A(\mathcal{L}_{k+1})\) must hold, for all \(k \in \{1, \ldots, |\mathcal{L}(\Pi)| - 1\}\).

Symmetry can also be present when adding the \(k\)th customer to a path, with \(k > 1\). For example, in Figure 3, two multilabels with paths starting from, possibly different, satellites \(s_i\) and \(s_j\) are given. Before performing the extension indicated by the dashed line, the multilabels...
represent different tour-trees. However, identical multilabels are obtained when adding customer 4 to \( L_2 \) and customer 3 to \( L'_1 \). To prevent this symmetry in the multilabels, we only allow the extension of a path if all subsequent paths visit at most one customer.

**Rule 2.** Path \( L_k \) is only extended if paths \( L'_j \) visit at most one customer, that is, \( L'(m_j) \leq 1 \), for all \( j \in \{ k+1, \ldots, L'(II) \} \).

This rule does not eliminate the possibility of having multilabels with paths visiting multiple customers. It only restricts the order in which they are being created; that is, we obtain a multilabel with paths visiting multiple customers by first extending the first path to multiple customers, then the second path, and so on. Once we have extended the second path, Rule 2 does not allow further extension of the first path. However, there will be other multilabels with the desired extension of the first path. For example, in Figure 3, we cannot perform the extension indicated by the dashed line of multilabel \( L' \); the desired multilabel can be obtained from multilabel \( L \).

The last symmetry rule concerns the orientation of customers in paths that can no longer be extended. In Figure 4, paths \( L_1 \) and \( L'_1 \) can no longer be extended once the dashed extensions of paths \( L_2 \) and \( L'_2 \) are performed (Rule 2). Both \( L_1 \) and \( L'_1 \) visit the same sequence of customers, but in the opposite order; that is, they represent the same SE route.

As we have no time windows, it holds that if a path is feasible, the reverse of that path 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.

**Rule 3.** Path \( L_k \) is only extended if \( a(L'_1) \times \zeta(L') > 1 \) for all \( j \in \{ 1, \ldots, k-1 \} \).

### 4.3. Two-Echelon One-Path Algorithm

To establish a benchmark for the multilabel 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. In essence, any tour-tree can be represented as a path in graph \( G \). As per example, the tour-tree \( (iv, (d, d, 3, 4, d) ), (c, c, 2, c, d), iv ) \) in Figure 1, is uniquely defined by the path \( (iv, d, 3, 4, d, c, 2, c, d, iv) \). The 2E-1P pricing algorithm constructs these paths iteratively, by appending vertices one-by-one to a partial path starting at a depot, until a complete path is obtained that has a one-to-one correspondence to a tour-tree.

The 2E-1P algorithm is inspired by one of the algorithms proposed by Dellaert et al. (2019). In contrast to the multilabel algorithm (Section 4.2), which generates elementary paths where customers can be visited at most once, in the 2E-1P algorithm we relax this requirement and allow \( ng \)-paths (Baldacci, Mingozzi, and Roberti 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 \( L \) in the 2E-1P algorithm is given in Table 2.

One of the attributes of the 2E-1P label is the cumulative joint density function \( F \) of the customer demands in the current SE route. This attribute is used to verify compliance with the capacity constraint. We obtain \( F \) by convoluting the cumulative density functions (CDF) of the demands of the customers in the current SE route. 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. We denote the CDF of customer demand \( \xi_i \), \( i \in C \), and its convolution with \( \psi(F) \) by \( F_i \) and \( (\psi(F) * F_i) \), respectively.

In each iteration of the 2E-1P algorithm, we randomly select the next label \( \mathcal{L} \) to be extended and iteratively consider a node \( i \in V \) for extension. An extension is feasible if one of the following three conditions holds:

- **Condition 1.** \( i \in C \land \mathcal{L} \notin \mathcal{L}(r) \land \{ \mathcal{L}(n) \in C \land \{ \mathcal{L}(n) \in S \land \mathcal{L}(p) \in S \cup D \} \land (\psi(F) * F_i) (Q^{SE}) \geq \eta \),

- **Condition 2.** \( i \in S \land \mathcal{L}(n) \in D \land \mathcal{L}(n) \in C \land \mathcal{L}(s) = i \lor \{ \mathcal{L}(n) \in S \land \mathcal{L}(p) \in C \land i \in \mathcal{L}(U) \} \),

- **Condition 3.** \( i \in D \land \mathcal{L}(d) = i \lor \{ \mathcal{L}(n) \in S \land \mathcal{L}(p) \in C \} \).

Condition (1) establishes that, when node \( i \) is a customer, extending to node \( i \) is feasible if node \( i \) does not belong to the ng-set; further, the last node visited is a customer or the last node visited is a satellite and the second to last node visited is a satellite or depot; and, finally, visiting node \( i \) does not violate the chance constraint. Condition (2) specifies that, when node \( i \) is a satellite, the extension to node \( i \) is feasible if the last node visited is a depot; or, the last node visited is a customer and the last visited satellite is satellite \( i \); or, the last node visited is a satellite, the second to last node visited is a customer, and node \( i \) is in the set of usable satellites. Finally, condition (3) states that, when node \( i \) is a depot, the extension to node \( i \) is allowed if node \( i \) is the starting depot of the tour-tree and the last SE route in the tour-tree is finished, that is, the last node visited is a satellite and the second to last node visited is a customer.

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

\[
\begin{align*}
\mathcal{L}'(d) &= \mathcal{L}(d), \\
\mathcal{L}'(s) &= \begin{cases} i, & \text{if } i \in S, \\
\mathcal{L}(s), & \text{otherwise}, \end{cases} \\
\mathcal{L}'(n) &= i, \\
\mathcal{L}'(p) &= \mathcal{L}(n), \\
\mathcal{L}'(\Gamma) &= \begin{cases} (\mathcal{L}(\Gamma) \cap N_I) \cup \{i\}, & \text{if } i \in C, \\
\mathcal{L}(\Gamma), & \text{otherwise}, \end{cases} \\
\mathcal{L}'(U) &= \begin{cases} (\mathcal{L}(U) \setminus \{\mathcal{L}(s)\}), & \text{if } i \in S, \mathcal{L}(s) \neq i, \\
\mathcal{L}(U), & \text{otherwise}, \end{cases} \\
\mathcal{L}'(\phi) &= \begin{cases} \mathcal{L}(\phi) + C_{\mathcal{L}(n),i} - \lambda, & \text{if } i \in C, \\
\mathcal{L}(\phi) + C_{\mathcal{L}(n),i}, & \text{otherwise}, \end{cases} \\
\mathcal{L}'(r) &= \begin{cases} \mathcal{L}(r) + 1, & \text{if } i \in S, \mathcal{L}(n) \in C, \\
\mathcal{L}(r), & \text{otherwise}, \end{cases} \\
\mathcal{L}'(\bar{\mu}) &= \begin{cases} \mathcal{L}(\bar{\mu}) + \mu, & \text{if } i \in C, \\
0, & \text{otherwise}, \end{cases} \\
\mathcal{L}'(\bar{\sigma}^2) &= \begin{cases} \mathcal{L}(\bar{\sigma}^2) + \sigma^2, & \text{if } i \in C, \\
0, & \text{otherwise}, \end{cases} \\
\mathcal{L}'(F) &= \begin{cases} F_i, & \text{if } i \in C \land \mathcal{L}(n) \in S, \\
\mathcal{L}(F) * F_i, & \text{otherwise}. \end{cases}
\end{align*}
\]

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

---

**Figure 4.** Two Multilabels Containing the Same Customers in the First Paths but in Opposite Order

\[
\begin{align*}
\mathcal{L}_1: & \quad s_i \quad \rightarrow \quad 1 \quad \rightarrow \quad 2 \\
\mathcal{L}_2: & \quad s_j \quad \rightarrow \quad 3 \quad \rightarrow \quad 4 \\
\mathcal{L}'_1: & \quad s_i \quad \rightarrow \quad 2 \quad \rightarrow \quad 1 \\
\mathcal{L}'_2: & \quad s_j \quad \rightarrow \quad 3 \quad \rightarrow \quad 4
\end{align*}
\]
Definition 5 (Exact Dominance). Label \( \mathcal{L}_1 \) dominates label \( \mathcal{L}_2 \) if and only if

(i) \( \mathcal{L}_1(d) = \mathcal{L}_2(d) \),
(ii) \( \mathcal{L}_1(s) = \mathcal{L}_2(s) \),
(iii) \( \mathcal{L}_1(n) = \mathcal{L}_2(n) \),
(iv) \( \mathcal{L}_1(f) \subseteq \mathcal{L}_2(f) \),
(v) \( \mathcal{L}_2(U) \subseteq \mathcal{L}_1(U) \),
(vi) \( \mathcal{L}_1(\phi) \leq \mathcal{L}_2(\phi) \),
(vii) \( \mathcal{L}_1(r) \leq \mathcal{L}_2(r) \),
(viii) \( \mathcal{L}_1(F(x)) \leq \mathcal{L}_2(F(x)) \quad \forall x \in \{0, \ldots, Q^{SE}\} \).

Definition 5 states that if \( \mathcal{L}_1 \) and \( \mathcal{L}_2 \) represent two partial tour-trees with the same last node visited, extending \( \mathcal{L}_1 \) 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 node. Condition (iv) requires that the set of customers to which \( \mathcal{L}_1 \) cannot be extended is a subset of the set of customers to which \( \mathcal{L}_2 \) cannot be extended. Condition (v) ensures that the set of satellites that can be visited from \( \mathcal{L}_1 \) contains the satellites that can be visited from \( \mathcal{L}_2 \). Condition (vi) assures that \( \mathcal{L}_1 \) has a reduced cost that is at most equal to the reduced cost of \( \mathcal{L}_2 \). Condition (vii) requires that \( \mathcal{L}_1 \) contains at most the same number of SE routes as \( \mathcal{L}_2 \), which given that 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 \( \mathcal{L}_1 \) has a lower or equal value than the CDF of \( \mathcal{L}_2 \) for all values of \( x \in \{0, \ldots, Q^{SE}\} \).

Comparing CDFs for all \( x \in \{0, \ldots, Q^{SE}\} \) 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 \( \mathcal{L}_1 \) dominates label \( \mathcal{L}_2 \) if

(i)–(vii),
(ix) \( \mathcal{L}_1(\bar{m}) \leq \mathcal{L}_2(\bar{m}) \),
(x) \( \mathcal{L}_1(\bar{a}^2) \leq \mathcal{L}_2(\bar{a}^2) \),

where one of the requirements (iv)–(vii) or (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 2E-1P algorithm and the heuristic dominance rule, we solve the pricing problem again with the exact dominance rule.

4.3.1. 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 \( \Gamma \) to labels with last node \( n \) and a smaller ng-set \( \Gamma' \) such that \( \Gamma' \subset \Gamma \).

4.4. Completion Bounds

In both pricing algorithms, we control the growth of the labels with completion bounds. For each label, we compute the lowest attainable reduced cost (completion bound) by considering all possible extensions from that label. If this bound is nonnegative, the label can be safely discarded because the goal is to find labels (tour-trees) with negative reduced cost. 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 and \( q \) expressed in terms of expected customer demand. The cost of an edge \( (i,j) \in E \) equals

\[
d_{ij} = \begin{cases} c_{ij} - \lambda_j, & \text{if } j \in C, \\ \ell_{ij}, & \text{otherwise}, \end{cases}
\]

where \( \lambda_j, j \in C \), are the dual values obtained from Constraint (2). Visiting a node \( i \in C \) consumes \( \mu_i \) units of \( q \). Each time the pricing algorithm is invoked, the RCSP-bounds are precomputed. In this way, the RCSP bounds can be retrieved in constant time during the labeling procedures. To formulate the completion bounds for the labels in each of the pricing algorithms, some additional notation is required. Let \( \mu(C') = \sum_{i \in C'} \mu_i \) denote the aggregated mean demand of the customers in set \( C' \subset C \) and let \( \bar{\mu} = \max_{C' \subset C} \{ \mu(C') : \mathbb{P} (\sum_{i \in C'} \xi_i \leq Q^{SE}) \geq \eta \} \) be the maximum value of \( \mu(C') \) for which it holds that there exists a customer combination \( C' \subset C \) that does not violate the chance constraint. In Section 5.3, the process of obtaining \( \bar{\mu} \) is described.

4.4.1. Multilabel Algorithm. To derive the completion bound \( CB(\mathcal{C}) \) on multilabel \( \mathcal{C} \), we first compute the RCSP-bound for each path \( \mathcal{C}_k, k \in \{1, \ldots, |\mathcal{C}(\Pi)|\} \). Next, we combine the RCSP-bounds of the paths in a multilabel to evaluate the completion bound on the multilabel. Specifically, in Equation (4), we substitute the costs of finishing a path with its RCSP-bound, that is, 

\[
CB(\mathcal{C}) = \mathcal{C}(\phi) + \sum_{k=1}^{\mathcal{C}(\Pi)} \psi(\mathcal{C}(n_k), \mathcal{C}(n_k), \bar{\mu} - \mathcal{C}(\bar{\mu}_k)) - \delta,
\]
with $\tilde{\mu}$ as the initial value for $\mu$. Suppose now that path $\mathcal{L}_k$ has a positive completion bound, specifically, \[
\mathcal{L}(\phi_k) + \psi(\mathcal{L}(n_k), \mathcal{L}(\pi_k), \tilde{\mu} - \mathcal{L}(\tilde{\mu}_k)) > 0,
\]
for some $k \in \{1, \ldots, |\mathcal{L}|(\Pi)\}$. Creating a multilabel without path $\mathcal{L}_k$ would result in a multilabel with lower reduced cost. Therefore, if any of the paths in a multilabel have a positive completion bound, the multilabel is eliminated. In this way, nonpromising labels can be discarded at an early stage of the pricing algorithm.

4.4.2. 2E-1P Algorithm. The completion bound $CB(\psi)$ for a label $\psi$ in the 2E-1P algorithm can be derived in a similar way as the completion bound on a multilabel, but using the remaining FE vehicle capacity rather than the remaining SE vehicle capacity. Through $\tilde{\mu}$ and the assumption that each FE route can supply at most $\theta$ SE routes, we derive that an FE vehicle can satisfy a total mean demand of at most $\theta\tilde{\mu}$. The completion bound on a label $\psi$ can now be computed as follows:
\[
CB(\psi) = \mathcal{L}(\psi) + \psi(\mathcal{L}(n), \mathcal{L}(d), \theta\tilde{\mu} - \mathcal{L}(\tilde{\mu}) - \mathcal{L}(\psi)) - \delta.
\]

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 consists in computing the probability distributions of the customers in an SE route (customer combination) and verifying feasibility from the convoluted distribution (Section 5.1). Alternatively, instead of computing this probability exactly, we can approximate it with statistical inference tests (Section 5.2). The benefit of using statistical inference tests 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 infrequently 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 nonnegative integer variables $X$ and $Y$, the distribution of the sum $Z = X + Y$ can be obtained by
\[
P(Z = z) = \sum_{k=0}^{z} P(X = k)P(Y = z - k).
\]

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, with $z \in \{0, \ldots, Q^{SE}\}$, and compute the probability that the total demand of the customers does not exceed the SE vehicle capacity. The complexity of this algorithm is $O(|C'| \cdot (Q^{SE})^2)$.

5.2. Statistical Inference Tests with Monte Carlo Sampling

In statistical inference, Monte Carlo sampling is used to estimate the probability $p(C') = P(\sum_{i \in C'} d_i \leq Q^{SE})$ that customer combination $C' \subset C$ satisfies the chance constraint. The approach consists in iteratively evaluating demand scenarios and computing a confidence interval $[a, b]$ around the estimated feasibility probability, that is, $a \leq \hat{p}(C') \leq b$ (Florio et al. 2021). We use the method by Agresti and Coull (1998) to estimate this interval.

We consider five standard deviations when defining the interval, which translates to a probability of 99.99994% that $[a, b]$ contains $p(C')$. 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 and evaluated to narrow the confidence interval. Algorithm 1 summarizes the procedure.

Algorithm 1 (Statistical Inference Test with Monte Carlo Sampling)

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

It may happen that Algorithm 1 terminates with an inconclusive result. In this case, we apply the following two-phase procedure for checking feasibility. First, we consider the statistical inference test with Monte Carlo sampling. If this test is inconclusive, the action to be taken depends on the demand distributions of the customers in $C' \subset C$. In the case of independent demand distributions, we convolute the demand distributions as explained in Section 5.1 and compute the exact value of $p(C')$. Otherwise, we resort to computing
the single point estimate \( \hat{\rho}(C') = \frac{x}{N} \) as our best guess for \( \rho(C') \) and derive (in)feasibility from it.

### 5.3. Feasibility Bounds

In this section, we develop the concept of feasibility bounds through which we can determine in constant time whether an SE route satisfies the probabilistic capacity constraint. Feasibility bounds rely on easy-to-compute properties of an SE route. One example of such a property is the sum of expected customer demands \( \mu(C') \) computed over the subset of customers \( C' \subset C \) visited in a (partial) SE route, that is, \( \mu(C') = \sum_{i \in C'} \mu_i \). If we can derive two bounds, say \( \bar{\mu} \) and \( \tilde{\mu} \), such that all customer combinations \( \{ C' \subset C \mid \mu(C') < \bar{\mu} \} \) satisfy the capacity constraint and all customer combinations \( \{ C' \subset C \mid \mu(C') > \tilde{\mu} \} \) violate the capacity constraint, then it suffices to only invoke the more computationally expensive feasibility checks for SE routes visiting a set of customers \( C' \) with \( \bar{\mu} \leq \mu(C') \leq \tilde{\mu} \). Although the lower and upper bounds depend on \( \eta \), for brevity’s sake, we omitted the index \( \eta \).

Using the upper bound \( \tilde{\mu} \), we redefine the right-hand side of Constraint (3) as follows:

\[
\gamma = \left[ \frac{\sum_{i \in C} \mu_i}{\theta \tilde{\mu}} \right],
\]

where \( \theta \) is the maximum number of SE vehicles that can be supplied from one FE vehicle, as defined in Section 4.2. The validity of this inequality follows from the observation that the fraction \( \sum_{i \in C} \mu_i / \tilde{\mu} \) constitutes a bound on the minimum number of SE routes in any feasible solution.

In the remainder of this section, we explain how to calculate \( \bar{\mu} \) and \( \tilde{\mu} \). These bounds are computed without assuming demands to be independent. Moreover, we precompute these bounds prior to the start of the column generation algorithm, as they are derived directly from the finite set of customers \( C \). Finally, as elaborated at the end of this section, instead of the total expected demand, we can also employ alternative properties such as the total demand variance to derive a different (complementary) set of bounds.

The procedure to calculate bounds \( \bar{\mu} \) and \( \tilde{\mu} \) is outlined in Algorithm 2. To obtain the value for the lower bound \( \bar{\mu} \), 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 \( \mu' \) by one and continue to do so until we reach a value \( \mu' \) for which there exists an infeasible customer combination \( C' \) with \( \mu(C') = \mu' \). We set the lower bound \( \bar{\mu} = \mu' \). Next, we initialize \( \mu' \) with the total mean demand of the customers in the instance, and decrease \( \mu' \) until we obtain a value \( \mu' \) for which there exists a customer combination \( C' \), \( \mu(C') = \mu' \), that satisfies the chance constraint, and set the upper bound \( \tilde{\mu} = \mu' \).

#### Algorithm 2 (Feasibility Bounds on the Expected Customer Demand)

1: **Input**: Set of customers \( C \)
2: \( \mu', \mu, \tilde{\mu} = 0, -1, -1 \)
3: while \( \tilde{\mu} = -1 \) do
   4: if \( \text{findInfeasibleCC}(\mu') \) then
      5: \( \mu = \mu' \)
      6: \( \mu' = \mu' + 1 \)
      7: \( \mu' = \sum_{i \in C} \mu_i \)
   8: while \( \bar{\mu} = -1 \) do
      9: if \( \text{findFeasibleCC}(\mu') \) then
         10: \( \tilde{\mu} = \mu' \)
         11: \( \bar{\mu} = \mu' - 1 \)

The problem of identifying (in)feasible customer combinations for a given value of \( \mu' \) (Lines 4 and 11 in Algorithm 2) can be formulated as an integer programming problem. Let \( \Omega \) be a set of scenarios with demand realization \( x_i^\omega \) for customer \( i \in C \) in scenario \( \omega \in \Omega \). Let \( x_i \) be a binary decision variable that takes the value of one if customer \( i \in C \) is selected. Binary decision variables \( y_\omega \) are set to one if the selected customers violate the vehicle capacity \( Q^SE \) in scenario \( \omega \in \Omega \), and zero otherwise. The following model is used to determine whether there exists an infeasible customer combination \( C' \subset C \) with \( \mu(C') = \mu' \):

\[
\begin{align*}
\text{minimize} & \quad 1 - \frac{1}{|\Omega|} \sum_{\omega \in \Omega} y_\omega, \\
\text{subject to} & \quad \sum_{i \in C} x_i^\omega \geq Q^SE - M(1 - y_\omega), \quad \omega \in \Omega, \\
& \quad \sum_{i \in C} x_i = \mu', \\
& \quad x_i \in \{0, 1\}, \quad i \in C, \\
& \quad y_\omega \in \{0, 1\}, \quad \omega \in \Omega.
\end{align*}
\]

The objective is to minimize the single point estimate of the feasibility probability over all customer combinations \( C' \subset C \) with \( \mu(C') = \mu' \). If the objective value is smaller than \( \eta \), there exists a combination \( C' \) with \( \mu(C') = \mu' \) that violates the chance constraint. Constraints (6) link the \( x_i \) and \( y_\omega \) variables. If the total demand of the selected customers in scenario \( \omega \in \Omega \) exceeds the vehicle capacity \( y_\omega \) is set to 1. Constraint (7) ensures that we only consider customer combinations \( C' \subset C \) with \( \mu(C') = \mu' \). Constraints (8) and (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):

\[
\begin{align*}
\text{maximize} & \quad 1 - \frac{1}{|\Omega|} \sum_{\omega \in \Omega} y_\omega, \\
\text{subject to} & \quad \sum_{i \in C} x_i^\omega \leq Q^SE + My_\omega, \quad \omega \in \Omega,
\end{align*}
\]

Hence, when the objective value is larger than \( \eta \), there exists at least one feasible customer combination.
Luedtke (2014) proposes a decomposition approach to avoid the big M-constraints, which lead to weak lower bounds when solving the continuous relaxation. Canessa et al. (2019) show that the concept of infeasible irreducible subsystems can be used as a tool to solve binary chance constraint models, where one iteratively eliminates scenarios from the problem formulation until the problem becomes feasible. Although the integer programming formulation can be solved using the techniques from Luedtke (2014) and Canessa et al. (2019), we proceed with an efficient search technique that enables us to use the statistical inference tests. For a given value of $\mu^*$, we retrieve all customer combinations $C' \subset C$ with $\mu(C') = \mu^*$ by solving a perfect subset sum problem with dynamic programming and store the obtained customer combinations in set $C$. Next, we iterate over the customer combinations in $C$ and invoke the two-phase procedure to verify compliance with Constraint (1). When searching for a(n) (in)feasible customer combination in $C$, we terminate the search once we identify such a combination and omit evaluating any remaining customer combination in $C$.

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 $\mu \leq \mu(C') \leq \tilde{\mu}$. Further reduction of the size of this set is achieved by imposing additional bounds expressed in terms of the total variance of customer demands. The procedure for obtaining these additional bounds is similar to the procedure for obtaining the bounds on the expected demand and is summarized in Online Appendix B.

6. Computational Results

We test the algorithms on instances proposed by Dellaert et al. (2019), which are publicly available in the online appendix of Sluijk et al. (2022). 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 $Q^{FE} = 150$ and $Q^{SE} = 50$.

In Section 6.1, we present the results on instances with independent demand distributions. Correlated customer demands are considered in Section 6.2. Finally, in Section 6.3, we address the efficiency of the statistical inference tests. All experiments are performed on a single thread of an Intel Xeon E5-2696 (2.4 GHz) CPU with 18 GB of memory. The (R)MP is solved using IBM CPLEX version 12.10. Finally, a time limit of two hours, including preprocessing 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, because, 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. 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 with solving the stochastic formulation. Finally, in Section 6.1.5, we analyze the impact of different chance constraint levels on the computation time, objective value, and feasibility of SE routes.

6.1.1. Performance Comparison of Multilabel 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 “No.” “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 to MP by solving MP directly with the set of tour-trees generated by the pricing procedure. In terms of lower bounds, the bounds computed with the multilabel algorithm are at least as strong as the bounds computed with 2E-1P algorithm, because 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 Online Appendix C.

Using the multilabel algorithm, we solve the RMP to optimality for 52 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 three 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 multilabel 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 multilabel 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 multilabel algorithm (2E-1P algorithm). The graph shows that optimality gaps between 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%.

It is apparent from Table 3 that the computation times increase exponentially with the number of customers in the instance. Comparing the computation times of the two algorithms, we note that, on average, the multilabel 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 the dominance rules in the 2E-1P algorithm.

### 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 multilabel algorithm and with or without the feasibility bounds. In the latter case, we derive the feasibility bounds to obtain a value for $\bar{\mu}$, but do not use the feasibility bounds when verifying the chance constraint. In this way, the same truncated vehicle capacity $\bar{\mu}$ 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 3. Independent Demands: Summarized Results**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Multilabel</th>
<th></th>
<th></th>
<th>Two-echelon one-path</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No.</td>
<td>Gap (%)</td>
<td>Col.</td>
<td>CPU (s)</td>
<td>No.</td>
<td>Gap (%)</td>
</tr>
<tr>
<td>15 customers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(2,3)</td>
<td>5</td>
<td>3.64</td>
<td>427</td>
<td>3</td>
<td>5</td>
<td>4.50</td>
</tr>
<tr>
<td>(3,5)</td>
<td>5</td>
<td>4.70</td>
<td>682</td>
<td>6</td>
<td>5</td>
<td>6.05</td>
</tr>
<tr>
<td>(6,4)</td>
<td>5</td>
<td>3.40</td>
<td>742</td>
<td>3</td>
<td>5</td>
<td>3.61</td>
</tr>
<tr>
<td>Overall</td>
<td>15</td>
<td>3.91</td>
<td>617</td>
<td>4</td>
<td>15</td>
<td>4.72</td>
</tr>
<tr>
<td>30 customers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(2,3)</td>
<td>5</td>
<td>4.73</td>
<td>1,584</td>
<td>43</td>
<td>5</td>
<td>4.39</td>
</tr>
<tr>
<td>(3,5)</td>
<td>5</td>
<td>2.17</td>
<td>1,666</td>
<td>55</td>
<td>5</td>
<td>2.69</td>
</tr>
<tr>
<td>(6,4)</td>
<td>5</td>
<td>3.65</td>
<td>1,649</td>
<td>24</td>
<td>5</td>
<td>3.11</td>
</tr>
<tr>
<td>Overall</td>
<td>15</td>
<td>3.52</td>
<td>1,633</td>
<td>41</td>
<td>15</td>
<td>3.39</td>
</tr>
<tr>
<td>50 customers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(2,3)</td>
<td>5</td>
<td>2.79</td>
<td>3,190</td>
<td>871</td>
<td>5</td>
<td>2.79</td>
</tr>
<tr>
<td>(3,5)</td>
<td>5</td>
<td>4.19</td>
<td>3,194</td>
<td>632</td>
<td>5</td>
<td>3.07</td>
</tr>
<tr>
<td>(6,4)</td>
<td>5</td>
<td>3.23</td>
<td>3,338</td>
<td>205</td>
<td>5</td>
<td>3.20</td>
</tr>
<tr>
<td>Overall</td>
<td>15</td>
<td>3.40</td>
<td>3,241</td>
<td>567</td>
<td>15</td>
<td>3.02</td>
</tr>
<tr>
<td>75 customers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(2,3)</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>2.30</td>
</tr>
<tr>
<td>(3,5)</td>
<td>4</td>
<td>2.92</td>
<td>6,322</td>
<td>2,889</td>
<td>4</td>
<td>2.66</td>
</tr>
<tr>
<td>(6,4)</td>
<td>3</td>
<td>2.54</td>
<td>6,765</td>
<td>1,396</td>
<td>4</td>
<td>2.31</td>
</tr>
<tr>
<td>Overall</td>
<td>7</td>
<td>2.76</td>
<td>6,512</td>
<td>2,249</td>
<td>12</td>
<td>2.43</td>
</tr>
</tbody>
</table>

*aAverage is taken over the solved instances.*
6.1.4. Value of the Stochastic Formulation. As an alternative to solving the 2E-VRPSD, one could solve a simpler, nonstochastic 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:

\[
\sum_{i \in C} \mu_i \leq Q^{SE} - s,
\]

where parameter \( s, 0 \leq s < Q^{SE} \), 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 because 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, with both formulations for different values of \( \eta \) and \( s \), using the multilabel 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 \( \eta \), 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 (truncated SE vehicle capacity of 40), the probability of having insufficient vehicle capacity exceeds 5%; when 40% of the vehicle capacity is reserved (truncated SE vehicle capacity of 30), 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 \( \eta \) and \( s \), confirming the superiority of the stochastic formulation over TEVP.

6.1.5. Different Chance Constraint Levels. Figure 7 shows for different chance constraint levels the average

![Figure 5. (Color online) Comparison of Optimality Gaps](image-url)

### Table 4. Runtime (in Seconds) of Multilabel Algorithm With/Out Feasibility Bounds (FB)

| \(|C|\) | Without FB | With FB | Speed up factor\(^a\) |
|-------|------------|---------|----------------------|
| 15    | 3.5        | 3.7     | 0.89                 |
| 30    | 54.1       | 40.8    | 1.33                 |
| 50    | 938.7      | 576.2   | 1.57                 |
| 75\(^a\) | 4,287.7    | 2,248.8 | 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.

### Table 5. Structural Characteristics of 2E-VRPSD Solutions

| \(|C|\) | \(|C| = 15\) | \(|C| = 30\) | \(|C| = 50\) | \(|C| = 75\) |
|-------|-------------|-------------|-------------|-------------|
| No. tour-trees | 2.9 | 4.9 | 7.5 | 11.4 |
| No. SE routes | 7.3 | 13.5 | 21.8 | 32.6 |
| No. SE routes in a tour-tree | 2.5 | 2.8 | 2.9 | 2.9 |
| No. customers in a tour-tree | 5.3 | 6.2 | 6.7 | 6.6 |
| No. customers in an SE route | 2.1 | 2.2 | 2.3 | 2.3 |
computation time, objective value, and feasibility probabilities of the lower bound solutions obtained on the instances with 30 and 50 customers. The plot on the left shows no clear relationship between the chance constraint levels and the computation times. The remaining two figures show an upward trend in the average objective value and feasibility probabilities of the SE routes as the chance constraint level increases. Although a higher chance constraint level reduces the likelihood of not being able to serve a customer in an SE route, it comes at the cost of a higher objective value. Moreover, the rightmost plot shows that the average feasibility probability of the SE routes in the solutions is often much higher than the chance constraint level. In Section 6.1.3, we showed that only a few customers are visited in each SE route in the solutions to the instances. Hence, adding a customer to or removing a customer from an SE route has substantial impact on its feasibility probability, leading to higher feasibility probabilities than required. In practice, when deciding on the value for the chance constraint level, one should carefully tradeoff the negative impact on the cost and the positive impact on the feasibility of the SE routes.

6.2. Correlated Demands

Thus far, only problem instances with independent customer demands are considered. In practice, however, customer demands are often correlated, because of a wide range of external factors that may induce correlations, such as weather, events, and sales (Gendreau, Jabali, and Rei 2016). 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 with negatively correlated demands, 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 violating the chance constraint.

To verify these assumptions, we conduct a number of experiments on 2E-VRPSD instances with correlated customer demand. In these experiments, we examine the impact of different correlation levels.

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) |
| Number of depots | 1.8              | 2.2               | 2.8               | 1.8 | 3.0 | 3.8 | 2.0 | 3.0 | 3.3 |
| Number of satellites | 2.2           | 2.6               | 3.0               | 2.4 | 4.6 | 3.8 | 3.0 | 5.0 | 4.0 |

\(a\)Number of depots and satellites available.

Figure 6. (Color online) Pareto Frontiers Showing the Value of the Stochastic Formulation
(weak and strong correlations), as well as the influence of two types of correlation (positive and mixed correlations). To generate instances with mixed correlations, we arbitrarily divide the set of customers into two groups. Customers within the same group have positively correlated demands, whereas customers from different groups have negatively correlated demands (Table 7). An example of the different correlations in an instance with 30 customers is given in Figure 8.

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 for each customer, we bypass this step and directly generate correlated demand scenarios as follows. For each customer \( i \in C \), we first generate independent Poisson distributed demand scenarios. To transform these independent demand scenarios into correlated demand scenarios, we multiply the demand scenarios by a random variable. To be precise, let \( d_{i}^{\omega} \) be the Poisson generated demand realization of customer \( i \in C \) in scenario \( \omega \). Let \( \alpha_{\omega} \) be the correlation parameter in scenario \( \omega \), which follows a uniform distribution \( U(a, b) \) with mean equal to one, where \( a \) and \( b \) are user-defined parameters that control the strength of the correlation. The positively correlated demand scenarios are generated as follows:

\[
\xi_{i}^{\omega} = \alpha_{\omega} \cdot d_{i}^{\omega}.
\]  

(11)

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

\[
\xi_{i}^{\omega} = ((1 - \alpha_{\omega}) + 1) \cdot d_{i}^{\omega}.
\]  

(12)

By construction of the correlated demand scenarios, the expected customer demand \( \mu_{i} \) of a customer \( i \in C \), is the same in both sets of scenarios.

Before we proceed with an in-depth analysis of our benchmark instances, we illustrate the impact of ignoring correlations between customer demands on a small problem instance containing four customers with strong positively correlated demands (\( \rho = 0.7 \), \( a = 0.3 \), and \( b = 1.7 \). Figure 9 shows two solutions for this instance: the solution on the left is obtained using the independent demand scenarios, whereas the solution on the right is retrieved with the correlated demand scenarios. The corresponding travel distances are 22.92 and 27.16. As can be observed from the figure, the solutions are fundamentally different in terms of their total distance traveled, number of SE routes, and number of satellites visited in the FE route. However, more importantly, if we re-evaluate the solution on the left on the correlated demand scenarios, we observe that both SE routes violates the chance constraint level. In other words, the solution obtained with the independent demand scenarios is not feasible when the customer demands are correlated. Thus, to meet the required chance constraint level, it is important to take the correlations into account when constructing the routes.

Next, we investigate the impact of demand correlations on our solution approaches and the 2E-VRPSD solutions. For these experiments, we reuse the same

<table>
<thead>
<tr>
<th>Group A</th>
<th>Group B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive</td>
<td>Negative</td>
</tr>
<tr>
<td>Negative</td>
<td>Positive</td>
</tr>
</tbody>
</table>

Table 7. Mixed Correlation
problem instances from the previous subsection, and only change the customer demand distributions. We consider two correlation levels: weakly correlated ($\rho = 0.3$) and strongly correlated ($\rho = 0.7$) customer demands. The demand scenarios for these correlation levels are obtained by setting $(a, b) = (0.7, 1.3)$ and $(a, b) = (0.3, 1.7)$, respectively. Because the 2E-1P pricing algorithm cannot handle instances with correlated demands, we use the multilabel algorithm to solve the instances. Table 8 compares the results obtained

![Figure 8](https://example.com/image1)

**Figure 8.** (Color online) Pearson Correlation Coefficients Between Each Pair of Customers Obtained on One Instance for the Different Settings

![Figure 9](https://example.com/image2)

**Figure 9.** (Color online) Solutions to an Instance with Correlated Demands ($\eta = 0.95$ and $Q_{SE} = 50$)

<table>
<thead>
<tr>
<th>Solution ignoring correlations</th>
<th>Solution assuming correlations</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu_1 = 25$</td>
<td>$\mu_1 = 25$</td>
</tr>
<tr>
<td>$\mu_2 = 18$</td>
<td>$\mu_2 = 18$</td>
</tr>
<tr>
<td>$\mu_3 = 12$</td>
<td>$\mu_3 = 12$</td>
</tr>
<tr>
<td>$\mu_4 = 16$</td>
<td>$\mu_4 = 16$</td>
</tr>
</tbody>
</table>

Note: Ignoring the correlations when solving the instance results in a solution with SE routes violating the chance constraints.
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 Online Appendix D.

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 demonstrate the importance of incorporating demand correlations, we solve the instances with independent Poisson generated demand, compute the feasibility probabilities of the corresponding routes under the different correlated demand scenarios, and derive the percentage of SE routes with a feasibility probability less than the chance constraint level for each correlation setting (infeasibility levels). Table 9 shows that the infeasibility levels range from 27.68% up to 88.83%. To conclude, it is important to adequately capture any dependence between customer demands, which can be achieved with the multilabel algorithm and statistical inference tests.

### Table 8. Correlated Demand: Summarized Results

<table>
<thead>
<tr>
<th>Correlated setting</th>
<th>Objective value</th>
<th>Number of SE routes</th>
<th>CPU (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \rho = 0.3 )</td>
<td>9.41</td>
<td>12.51</td>
<td>-28.94</td>
</tr>
<tr>
<td>( \rho = 0.7 )</td>
<td>41.82</td>
<td>55.78</td>
<td>-54.83</td>
</tr>
<tr>
<td>Mixed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \rho = 0.3 )</td>
<td>2.90</td>
<td>2.21</td>
<td>-8.47</td>
</tr>
<tr>
<td>( \rho = 0.7 )</td>
<td>10.78</td>
<td>9.86</td>
<td>16.03</td>
</tr>
</tbody>
</table>

### 6.3. Efficiency of the Statistical Inference Tests

In this section, we compare Algorithm 1 with the approach proposed by Dinh, Fukasawa, and Luedtke (2018) to verify route feasibility. Recall that the statistical inference test can be applied to any customer demand distribution that can be sampled from. The approach in Dinh, Fukasawa, and Luedtke (2018), from hereon referred to as the quantile test, requires that one can compute a quantile of the sum of customer demands for any subset of customers. With a scenario-based demand distribution, the quantile test for a subset of customers \( C' \subset C \) is performed by deriving the distribution of the total demand of the customers in \( C' \) from scenarios, computing the \((1 - \eta)\) th quantile and comparing it to \( Q^{SE} \).

We compare both methods on an instance with 100 customers. The scenarios are generated with the procedure described in Section 6.2 with \( \alpha \) uniformly distributed on \([0.7, 1.3]\). We set \( Q^{SE} = 100 \) and perform the tests on 10,000 customer combinations \( C' \subset C \) with \( 20 \leq \mu(C') \leq 85 \). Figure 10 summarizes the results. The left plot depicts the average percentage deviation between the correlations in the true correlation matrix, based on 10\(^6\) scenarios, and the correlations in the correlation matrix obtained on fewer scenarios. Although Dinh, Fukasawa, and Luedtke (2018) used 200 scenarios in their computational experiments, we note that at least 2,000 scenarios are needed for a reasonable approximation of the correlation structure of random demands. The plot on the right shows the number of scenarios required for asserting feasibility within four standard deviations, which translates to a probability of 99.9936% that the conclusion is correct. Each bar shows the number of customer combinations for which (in)feasibility is confirmed with the indicated number of scenarios. For about 9% of the customer combinations, no conclusion was reached within 5,000 scenarios and we asserted (in)feasibility from the single point estimate. The number of infeasible and feasible customer combinations are represented by the the dark and light gray bars, respectively. We observe that fewer scenarios are required to confirm infeasibility as opposed to feasibility, which is due to the chance-constraint level enforced. The statistical inference test requires, on average, only 1,040 scenarios to determine feasibility, whereas the quantile test always requires evaluating the entire set of scenarios.

### 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 con-
sider two solution procedures based on column gener-
ation. We propose the multilabel 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 tests to speed
up feasibility check operations. When these are incon-
clusive, we consider the convolution of probability
distributions in the case of independent demand dis-
tributions and resort to single point estimates when
demands are correlated. To reduce the computational
efforts on validating the chance constraint, we intro-
duce the notion of feasibility bounds and apply them
to the sum of the means and variances of the cus-
tomer demands.

Computational experiments are performed on instan-
ces with independent demands and on instances with
correlated demands. We compute linear bounds on
instances with independent demands and up to 75 cus-
tomers within a time limit of two hours. Regardless of
the labeling procedure, the average optimality gap is
less than 4%. Moreover, we show that with the use of
feasibility bounds, the labeling algorithms’ runtimes are
reduced significantly.

We demonstrate that solving the 2E-VRPSD always
results in better solutions in terms of objective value
and feasibility of SE routes compared with solving the
expected value 2E-VRP with a 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
when deciding on the value for the chance constraint
level, one should carefully tradeoff the negative
impact on the cost and the positive impact on the fea-
sibility of the SE routes.

The multilabel algorithm, statistical inference tests
and feasibility bounds do not require demands to be
statistically independent. We demonstrate that ignoring
demand correlations when constructing route plans
leads to SE routes that violate the chance constraint
when evaluated with the correlated demand scenarios.
The percentage of infeasible SE routes depends on the
type of correlation and varies from 28% up to 89%.

Future research on the chance-constrained two-echelon
vehicle routing problem with stochastic demands is rec-
ommended 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.

Future research on the chance-constrained 2E-VRPSD
may investigate problem extensions that include addi-
tional real-life aspects such as time windows at custom-
ners or variants that allow SE vehicles to receive freight
from multiple FE vehicles. From the methodological
side, another promising research direction is the
exploration of additional acceleration techniques to
improve the performance of the pricing algorithms,
for example, bidirectional labeling and arc fixing, and how to adapt those techniques to handle probabilistic resource constraints.

Acknowledgments
The authors thank the editors and two anonymous referees for comments, which led to significant improvements in this paper.

References


