Fair stochastic vehicle routing with partial deliveries

N. Sluijk, W. Rei, J. Kinable, M. Gendreau, T. Van Woensel

April 2023

Abstract

A common assumption in the models for the vehicle routing problem with stochastic demands is that all demands must be satisfied. This is achieved by including recourse actions in two-stage stochastic programming formulations or by ensuring with a high probability that all demand fits within the vehicle capacity (chance-constrained formulations). In this work, we relax the assumption of full demand satisfaction and allow partial deliveries. Practical applications of partial deliveries include humanitarian logistics and food rescue programs. To ensure a fair solution for all customers, we require that the minimum expected fill rate over all customers meets the target fill rate. We refer to the resulting problem as the fair stochastic vehicle routing problem with partial deliveries. We propose a model in which we account for uncertain customer demand by constructing routes such that the expected minimum fill rate is above a predefined threshold. To solve the problem, we develop a branch-price-and-cut algorithm capable of solving instances with up to 75 customers. Specifically, we propose problem-specific bounding techniques to enhance the performance of the solution methods for the pricing problem. Results show, among others, that with our proposed model, solutions are guaranteed to be feasible at only a marginal cost increase compared to a deterministic model with expected demands.

1 Introduction

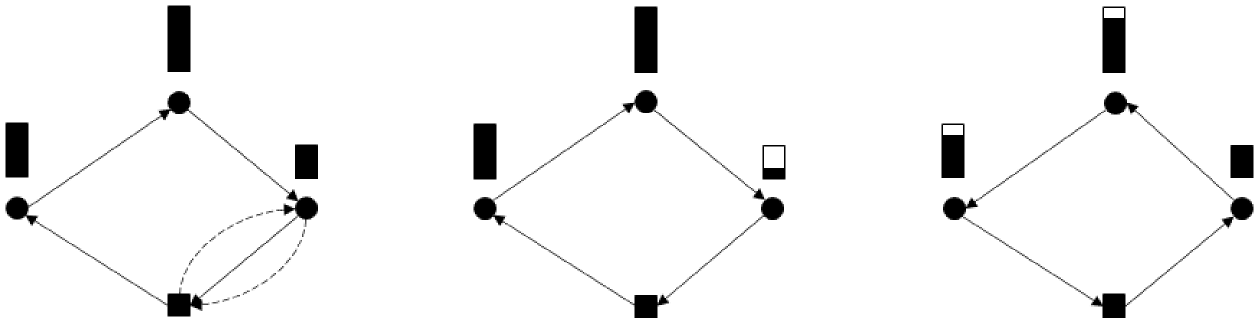
The capacitated vehicle routing problem (CVRP) aims at finding a set of feasible routes that minimizes the total routing costs subject to the constraint that each customer is visited exactly once. Each route should start at the depot, visit some customers, and return to the depot. A route is feasible if the previous structure holds and the total demand of the customers visited in the route does not exceed the vehicle capacity. Most work is performed on the deterministic CVRP, where all problem parameters are certain. However, in reality, some of the parameters may be uncertain. For example, when planning a route, we may not know the actual travel times, service times, and/or demand of each customer (Gendreau et al., 2016). This work focuses on the latter example of uncertainty and studies the vehicle routing problem with stochastic demands (VRPSD).

The VRPSD is often formulated using two-stage stochastic programming with recourse or chance-constrained programming. The former formulation involves two decision stages. In the first stage, routes must be planned without knowing the true values of the customers' demands. In the second stage, we observe a customer's demand upon arrival at their location, and recourse actions may be triggered by a pre-determined recourse policy (Oyola et al., 2018). The latter approach imposes chance constraints to ensure with high probability that the total demand of the customers visited in a route does not exceed the vehicle capacity (Dinh et al., 2018).

A common assumption in the VRPSD literature is that all demands must be met. In the two-stage stochastic formulation, this is achieved by performing recourse actions when the total demand of the customers in a route exceeds the vehicle capacity. All recourse policies considered in the literature involve a detour to the depot to restock a vehicle or a scheduled rendezvous of two vehicles to exchange capacity. However, such detours may not be feasible in practice due to time and/or safety restrictions, e.g., in humanitarian logistics, detours are undesirable due to bad road conditions and/or the high risk of barricades and raids along the routes (European Commission, 2022). In the chance-constrained formulation, no explicit actions are considered for dealing with excess demand. In other words, insufficient capacity may result in an unfair distribution of capacity over customers in a route.

One solution to these challenges is to relax this assumption and allow partial demand satisfaction. This relaxation is strongly motivated by several practical applications in which it is more important that all customers receive some products, rather than some customers receiving their demand and others not receiving anything. This is particularly relevant in commercial settings where spreading supply across multiple demand locations may result in higher revenue. Additionally, at non-profit organizations/operations such as food rescue programs and humanitarian logistics, the total demand frequently exceeds the supply, and partial deliveries are the only possibility (Rivera et al., 2023; Anuar et al., 2021). This relaxation introduces the need for additional decisions regarding the delivery quantities to each customer and, consequently, criteria to evaluate the quality of these decisions. In this paper, a distribution of capacity over customers is considered to be fair if the minimum utility over all customers exceeds a predefined threshold. Similar to, among others, Anaya-Arenas et al. (2018), Ibarra-Rojas and Silva-Soto (2021), and Nair et al. (2017), we use fill rates as a measure of utility, where the fill rate at a customer is equal to the proportion of demand supplied. As organizations frequently set a target fill rate, we consider a route feasible if, in expectation, the minimum fill rate over all customers visited in the route meets the target fill rate. This way, we ensure that the expected service level for each customer is at least equal to the target service level.

Constructing routes with a fair distribution of scarce resources along a route has mostly been studied under the assumption of deterministic demand (Khorsi et al., 2020; Eisenhandler and Tzur, 2019), except for Balcik et al. (2014). They propose a model and heuristic solution method for the stochastic multi-vehicle



(a) Two-stage stochastic programming formulation with recourse

(b) Chance-constrained formulation

(c) Fair stochastic vehicle routing with partial deliveries

Figure 1: Solutions obtained on a small instance with allocations displayed for one demand scenario in which the total demand exceeds the vehicle capacity.

sequential allocation problem to provide equitable service and minimize unused donations. Hence, routing costs are neglected when constructing the routes. This work considers the fair stochastic vehicle routing problem with partial deliveries. We aim to find a set of feasible routes that minimizes the total routing costs subject to the standard routing constraints and a constraint that ensures that the minimum expected fill rate over all customers exceeds a pre-defined target fill rate. The final allocation of capacity to a customer depends on its demand realization and is only decided upon when arriving at a customer and observing its demand. This assignment problem can be classified as a sequential resource allocation problem and the procedure in Lien et al. (2014) could be followed to decide on the allocation to each customer.

The differences between the solutions to the three approaches discussed so far - the two-stage stochastic formulation with recourse, the chance-constrained formulation, and our formulation for the fair stochastic vehicle routing problem with partial deliveries - are highlighted in Figure 1. The solutions show the planned routes and the proportion of demand fulfilled for a specific demand realization where the sum of demands exceeds the vehicle's capacity. Figure 1a shows that a recourse action is required with the two-stage stochastic programming approach to fulfill all demands. Conversely, the chance-constrained formulation (Figure 1b) does not entail any recourse action, resulting in partial satisfaction of the last customer's demand. Our formulation returns a solution (Figure 1c) with a greater minimum fill rate over all customers when compared to the chance-constrained formulation without using any recourse action.

In summary, our paper brings the following contributions:

- We propose a model for the fair stochastic vehicle routing problem with partial deliveries and account for uncertain customer demand by designing routes such that the expected minimum fill rate is above a predefined threshold.

- We develop a branch-price-and-cut algorithm capable of solving instances with up to 75 customers. Specifically, we propose problem-specific bounding techniques to enhance the performance of the solution methods for the pricing problem.

- We present an extensive set of numerical experiments to compare the results obtained with our proposed model to three alternative models. Results show, among others, that with our proposed model, solutions are guaranteed to be feasible at only a marginal cost increase (1.52% on average compared to the deterministic model).

The outline of the paper is as follows. In Section 2, the relevant literature is discussed, followed by a problem description in Section 3. In Section 4, we discuss several heuristic procedures for verifying the feasibility of a route concerning the fill rate constraint. The problem is solved with a branch-price-and-cut algorithm as described in Section 5. In Section 6, we introduce three alternative models to compare our results to in Section 7. Finally, in Section 8, we conclude this paper and provide suggestions for future research.

2 Literature review

The fair stochastic vehicle routing problem with partial deliveries builds on various existing vehicle routing problems. In Section 2.1, we review relevant literature on the vehicle routing problem with stochastic demands. Similarly, in Section 2.2, we review work on vehicle routing problems with fairness considerations.

2.1 Vehicle routing problem with stochastic demands

Three approaches have been considered for the VRPSD: stochastic programming with recourse, chance-constrained programming, and reoptimization. In stochastic programming with recourse, the problem is tackled in two stages. First, before any demand is revealed, a set of routes is constructed that minimizes the sum of routing costs and expected recourse costs. Then, upon execution of the routes, demand is revealed, and recourse actions may be taken. In the literature, different recourse policies have been considered, including the detour-to-depot policy (Gauvin et al., 2014; Jabali et al., 2014), optimal restocking policy (Salavati-Khoshghalb et al., 2019a), rule-based restocking policy (Salavati-Khoshghalb et al., 2019c), switch policy (Florio et al., 2022a), and other variants (Ak and Erera, 2007; Salavati-Khoshghalb et al., 2019b). All involve a detour to the depot or rendezvous between vehicles to exchange capacity.

In chance-constrained programming, chance constraints are included to ensure that the total demand of the customers visited in a route fits within the vehicle's capacity with high probability (Dinh et al., 2018; Noorizadegan and Chen, 2018; Sluijk et al., 2023). The resulting model does not include any recourse actions that should be taken when the vehicle's capacity is exceeded, nor does it consider how much the capacity is exceeded. In other words, exceeding it by one unit is considered as bad as exceeding it by half of the vehicle's capacity.

An alternative approach is to allow for complete reoptimization of the routes whenever new information is revealed, potentially resulting in a solution very different from the initial solution (Dror et al., 1989; Secomandi, 2001; Novoa and Storer, 2009). Some papers also consider partial reoptimization (Secomandi and Margot, 2009; Goodson et al., 2013).

The best-performing algorithms for the deterministic CVRP are based on a combination of column and cut generation, commonly known as the branch-price-and-cut algorithm (Costa et al., 2019). Additionally, various successful implementations of variants of the branch-price-and-cut algorithm on the VRPSD exist (Florio et al., 2020, 2022b; Hoogendoorn and Spliet, 2023). A significant advantage of branch-and-price is that the complexity related to the stochastic customer demand can be entirely handled inside the pricing problem. For a complete overview of the models and solution methods considered for stochastic vehicle routing problems, we refer to Oyola et al. (2018, 2017).

So far, we reviewed the literature on the VRPSD in which it is assumed that all demands must be met. Kyriakidis and Dimitrakos (2017) relax this assumption and include the possibility of partial demand satisfaction. Their cost function consists of three components: routing costs, detour-to-depot costs, and penalty costs incurred when a customer's demand is only partially satisfied. They propose an optimal routing policy and show with some numerical experiments that if the order of the customers is not fixed, it is possible to derive the optimal routing strategy by enumerating all possible sequences if the instance contains at most nine customers. Although penalizing partial deliveries will motivate efficient use of resources, it does not ensure a fair distribution of resources among customers.

2.2 Vehicle routing problems with fairness considerations

In recent years, fairness/equity issues have received more attention and can be found in a wide range of optimization models and application areas (Karsu and Morton, 2015; Chen and Hooker, 2022; Hooker, 2023). This also holds for the vehicle routing problem. Vidal et al. (2020) present an overview of a variety of equity criteria imposed on different problem attributes of the VRP, including workload balancing (Matl et al., 2018), service equity (Huang et al., 2012), and collaborative planning (Gansterer and Hartl, 2018). In the remainder of this section, we focus on service equity.

In disaster relief and food rescue programs, the total demand of stakeholders (e.g., affected areas or food banks) regularly exceeds the supply. This introduces the need for additional variables (next to routing decisions) on the quantities to deliver to each customer and, consequently, criteria to evaluate the quality of these decisions. Gralla et al. (2014) conducted a joint analysis survey among eighteen experienced humanitarian logisticians. Their results show that effectiveness is considered more important than efficiency, and practitioners prioritize more vulnerable communities and critical commodities while still acknowledging the needs of others. Furthermore, Huang et al. (2012) show that there is a significant difference between solutions that focus on the traditional commercial concern of efficiency (e.g., minimize cost) and solutions that focus on both equity and efficacy (quick and adequate response).

An overview of possible equity metrics can be found in Marsh and Schilling (1994). A common metric in the VRP literature is the fill rate at each customer (Anaya-Arenas et al., 2018; Khorsi et al., 2020; Lu et al., 2022). To evaluate fairness based on this metric, different criteria have been considered. For example, Ibarra-Rojas and Silva-Soto (2021) focus on an egalitarian distribution of resources and aim at maximizing the weighted sum of the minimum fill rate among all demand points and the sum of delivery times. For a comprehensive overview of the literature on multicriteria optimization in humanitarian aid, we refer to Gutjahr and Nolz (2016).

Most studies on vehicle routing problems involving partial deliveries and fairness considerations have assumed deterministic demand. However, in reality, demand is often stochastic. For instance, in the aftermath of a disaster, predicting the demand for goods and services in the affected areas may be difficult. Likewise, the number of people who may arrive at a food bank to receive a package of essential items is often unknown beforehand. Incorporating stochasticity into the models is important in bridging the gap between literature and practice. One work in that direction is by Balcik et al. (2014). They consider a multi-vehicle sequential allocation problem with stochastic demands. Their objective is two-fold: providing equitable service and minimizing unused donations. Hence, routing costs are neglected. Their solution method consists of clustering customers, sequencing customers, and allocating capacity to each customer. Alkaabneh et al. (2023) propose a two-stage stochastic programming formulation for integrated stochastic routing and resource allocation in the context of a mobile food pantry program. In the first stage, supply is pre-allocated to customers, and routes are constructed. At the beginning of the second stage, the true demand values are revealed, and the final allocations are determined. Here, some flexibility is taken into account by allowing any excess supply from previous customers to be (partially) used for the next customers. Instances with up to 35 customers are solved with an adaptive large neighborhood search algorithm.

This paper proposes a model and exact solution approach to obtain a set of routes that ensures a fair division of capacity over customers and minimizes the total routing cost. Each customer's demand is only revealed upon arrival at its location. Hence, when executing the routes, one has to decide on the allocation to the current customer without knowing the demands of the customers yet to visit. This problem is known as the sequential resource allocation problem, and we refer to Lien et al. (2014) for a procedure that could be followed to decide on the allocations to each customer.

3 Problem description

The problem is defined on a complete and directed graph $G = (V, A)$, with node set $V = \{0, \dots, n\}$ containing the depot (node 0) and customers ($V' = \{1, \dots, n\}$) and arc set $A = \{(i, j) : i, j \in V, i \neq j\}$. The cost of traversing arc $(i, j) \in A$ is equal to c_{ij}. We consider a homogeneous fleet of vehicles with a vehicle capacity of Q each. We assume that the customer demands are independent and identically distributed. Specifically, the demand of each customer $i \in V'$ is a discrete random variable $\xi_i \in \mathbb{N}$ with mean μ_i and $\mathbb{P}(\xi_i = 0) = 0$. We denote the target fill rate by F.

A route $r = (i_0, i_1, \dots, i_m, i_{m+1})$ with $i_0 = i_{m+1} = 0$ and $i_k \in V'$ for $k \in \{1, \dots, m\}$ is considered to be feasible if the expected minimum fill rate over all customers in the route $({}_r)$ meets the target fill rate. We refer to this fill rate as the collective fill rate and compute it with the following dynamic programming model:

$$v^r(k, q) = \begin{cases} \sum_{d=1}^{\infty} \mathbb{P}\big(\xi_{i_k} = d\big) \max\limits_{x \leq \min(d,q)} \min\left[\frac{x}{d}, v^r(k+1, q-x)\right] & \text{if } k = 1, \dots, m-1, \\ f_{i_k}(q) & \text{if } k = m, \end{cases} \tag{1}$$

where $f_{i_k}(q)$ is the expected fill rate at customer i_k with q units of capacity left:

$$f_{i_k}(q) = \sum_{d=1}^{\infty} \mathbb{P}\big(\xi_{i_k} = d\big) \frac{\min(q, d)}{d}. \tag{2}$$

To find the optimal allocation, and hence the highest possible value for the collective fill rate, we start at $k = m$ and recursively solve (1) for $k \in \{m, \dots, 1\}$ to obtain the collective fill rate value for route r, i.e., ${}_r = v^r(1, Q)$. The complexity of this algorithm is $\mathcal{O}(m \cdot U^d \cdot Q)$ with U^d equal to some large value such that for any $d > U^d$ it holds that $\mathbb{P}(U^d > d) \approx 0$, hence negligible. Note that the value of ${}_r$ depends on the order in which the customers are visited in the route. For example, suppose that we have an instance with two customers. Customer A requests 3, 5, or 7 units with probability 0.2, 0.6, and 0.2, respectively, and customer B requests eight units with probability 1. We set the capacity of the vehicle to $Q = 11$. Route $r = (0, \text{A}, \text{B}, 0)$ will give us a collective fill rate of

$$\begin{aligned} {}_r = v^r(1, 11) &= \sum_{d \in \{3,5,7\}} \mathbb{P}(\xi(A) = d) \max_{x \leq (d,11)} \min\left[\frac{x}{d}, v^r(2, 11-x)\right] \\ &= \mathbb{P}(\xi(A) = 3) \max_{x \leq 3} \min\left[\frac{x}{3}, \frac{11-x}{8}\right] \\ &\quad + \mathbb{P}(\xi(A) = 5) \max_{x \leq 5} \min\left[\frac{x}{5}, \frac{11-x}{8}\right] \end{aligned}$$

$$
\begin{aligned}
&+ \mathbb{P}(\xi(A) = 7) \max_{x \le 7} \min \left[\frac{x}{7}, \frac{11 - x}{8} \right] \\
&= \mathbb{P}(\xi(A) = 3) \min \left[\frac{3}{3}, \frac{8}{8} \right] + \mathbb{P}(\xi(A) = 5) \min \left[\frac{4}{5}, \frac{7}{8} \right] \\
&+ \mathbb{P}(\xi(A) = 7) \min \left[\frac{5}{7}, \frac{6}{8} \right] \\
&= 0.2 \cdot 1 + 0.6 \cdot 0.8 + 0.2 \cdot 0.71 = 0.82,
\end{aligned}
$$

whereas the reverse route $r' = (0, \text{B}, \text{A}, 0)$ will give us a collective fill rate of

$$
\begin{aligned}
{}_{r'} = v^{r'}(1, 11) &= \sum_{d \in \{8\}} \mathbb{P}(\xi(B) = d) \max_{x \le (d,q)} \min \left[\frac{x}{d}, v^r(2, 11 - x) \right] \\
&= \max_{x \le 8} \min \left[\frac{x}{8}, f_A(11 - x) \right] \\
&= \max_{x \le 8} \min \Bigg[\frac{x}{8}, \mathbb{P}(\xi(A) = 3) \frac{\min(11 - x, 3)}{3} \\
&\quad + \mathbb{P}(\xi(A) = 5) \frac{\min(11 - x, 5)}{5} + \mathbb{P}(\xi(A) = 7) \frac{\min(11 - x, 7)}{7} \Bigg] \\
&= \min \left[\frac{7}{8}, 0.2 \cdot \frac{3}{3} + 0.6 \cdot \frac{4}{5} + 0.2 \cdot \frac{4}{7} \right] \\
&= \min [0.88, 0.79] = 0.79.
\end{aligned}
$$

Hence, visiting customer A first results in a higher collective fill rate. However, if $Q = 6$, the collective fill rates for routes r and r' are 0.42 and 0.43, respectively, and visiting customer B first will result in a higher collective fill rate. The latter two observations imply that proposing any dominance rules on partial paths is intricate.

3.1 Mathematical model

We propose a set partitioning formulation for the problem. Let R denote the set of feasible routes. A route r is feasible if it starts and ends at the depot, visits each customer at most once, and has a collective fill rate ${}_r$ larger than or equal to the target fill rate F:

$$
{}_r \ge F. \tag{3}
$$

We denote the set of arcs traversed in route $r \in R$ by $A(r)$ and define its corresponding transportation cost as $c_r = \sum_{(i,j) \in A(r)} c_{ij}$. Parameter a_{ir} equals 1 if customer $i \in V'$ is visited in route r and 0 otherwise. Finally, binary decision variable y_r equals 1 if route r is selected and 0 otherwise. The problem is now formulated as follows:

$$
\begin{aligned}
&\min \sum_{r \in R} c_r y_r && && (4) \\
&\text{s.t.} \sum_{r \in R} a_{ir} y_r = 1, && i \in V', && (5) \\
&\qquad y_r \in \{0, 1\}, && r \in R. && (6)
\end{aligned}
$$

The objective is to minimize the total routing cost. Constraints (5) ensure that each customer is visited exactly once, and Constraints (6) set the domain of the variables. The proposed set partitioning formulation assumes a set of feasible routes. As this set of routes grows exponentially with the number of customers in the problem instance, we proceed with a linear relaxation of the formulation with an initial subset of feasible routes and use column generation to generate additional promising routes (Desaulniers et al., 2006). The advantage of this approach is that the challenge of checking feasibility with respect to the target fill rate can be handled entirely inside the algorithms used to generate new columns.

4 Heuristic feasibility checks

In Section 3, we showed that the collective fill rate of a route can be obtained with dynamic programming. As this procedure is computationally expensive, we propose two alternative methods for checking feasibility of customer sequences (routes) with respect to the fill rate constraint. Instead of computing the collective fill rate exactly, we could approximate it using statistical inference with Monte Carlo simulation as described in Section 4.1. In Section 4.2, we derive an upper bound on the sum of mean demand values of the customers in a route such that any route with an aggregated mean demand larger than this bound is infeasible.

4.1 Collective fill rate under perfect information

One way to quickly assess whether a route may meet the target fill rate is by approximating its collective fill rate through statistical inference tests with Monte Carlo sampling (Florio et al., 2021; Sluijk et al., 2023). First, we generate a scenario by sampling from the demand distributions of the customers in the route. Next, we assume that we know each customer's demand at the start of the route (perfect information), and use dynamic programming to decide on the allocation to each customer such that the minimum fill rate of the route is maximized. We iteratively evaluate demand scenarios and compute a confidence interval $[a, b]$ around the collective fill rate under perfect information following the procedure in Agresti and Coull (1998). The general outline of this procedure is given in Algorithm 1.

Algorithm 1 Collective fill rate under perfect information

```
input: set of customers V', maximum number of scenarios N, number of standard deviations κ, target fill rate F
s_1 ← 0, s_2 ← 0
for n = 1 to N do
    generate demand scenario [ξ_i]_{i∈V'}
    compute the collective fill rate f for demand scenario [ξ_i]_{i∈V'} with dynamic programming
    update s_1 ← s_1 + f and s_2 ← s_2 + f^2
    compute sample mean and variance as μ_n ← s_1/n and σ_n^2 ← s_2 − μ_n^2
    if μ_n + κ·sqrt(σ_n^2/n) < F then
        return infeasible
    if μ_n − κ·sqrt(σ_n^2/n) ≥ F then
        return feasible
return inconclusive
```

If $a > F$ ($b < F$), we conclude that the route is feasible (infeasible). If $F \in [a, b]$, no conclusions can be drawn, and additional scenarios must be evaluated to narrow the confidence interval. We consider at most N scenarios. If the conclusion is that the route is infeasible, we no longer need to compute its true collective fill rate, as this will always be less than the collective fill rate under perfect information. If the method returns inconclusive or feasible, we must compute its true collective fill rate to conclude (in)feasibility.

4.2 Upper bound on feasible mean

In branch-price-and-cut algorithms for vehicle routing problems, the vehicle capacity is often considered an upper bound on the aggregated mean demand in a route. That is, any route with a larger aggregated mean demand is considered to be infeasible. This is valid under the assumption that, on average, vehicles should be able to serve all demands without any recourse actions (Florio et al., 2022b). However, with partial deliveries, we may have feasible routes with an aggregated mean demand that exceeds the vehicle capacity. As valid inequalities, completion bounds, and other techniques within branch-price-and-cut algorithms typically require a limiting resource, we consider the following two-step procedure to derive an upper bound on the aggregated mean value and use that as the limiting resource within the methods mentioned above.

We derive a valid initial bound by considering a single artificial customer and maximizing its mean value subject to the constraint that its expected fill rate meets the target fill rate:

$$\bar{\mu} = \max \quad \left\{ \mu : \sum_{d=1}^{\infty} \mathbb{P}\big(\xi = d \big| \mu\big) \frac{\min(q, d)}{d} \geq F, \mu \in \mathbb{N} \right\}. \tag{7}$$

Next, to potentially achieve a stronger upper bound on the feasible mean, we investigate whether a feasible customer combination with an aggregated mean of $\bar{\mu}$ exists. If not, we decrease the value of $\bar{\mu}$ until a feasible combination is found. The level of uncertainty regarding customer demands increases with the size of the customer combinations. Therefore, we will only consider the smallest possible size for a given value of $\bar{\mu}$. To acquire this size, we solve the following problem:

$$l = \min \left\{ \sum_{i \in V'} x_i : \sum_{i \in V'} \mu_i x_i = \bar{\mu}, x_i \in \{0, 1\}, i \in V' \right\} \tag{8}$$

To find all customer subsets $S \subseteq V'$ with $\sum_{i \in S} \mu_i = \bar{\mu}$ and $|S| = l$, we solve a perfect subset sum problem (PSSP). We iterate over all customer subsets retrieved to verify whether a feasible sequence exists. First, we use the procedure outlined in Section 4.1. If this returns infeasible, we move on to the next customer subset. Otherwise, we proceed with computing its true collective fill rate, which requires iterating over all possible sequences of the customers in the subset. If at least one sequence is feasible, we conclude that a feasible sequence with mean $\bar{\mu}$ and size l exists, set the upper bound on the feasible mean to $\bar{\mu}$, and terminate our search. If none of the customer subsets return a feasible sequence, we conclude that no feasible subset with an aggregated mean of $\bar{\mu}$ exists and proceed with iteratively decreasing $\bar{\mu}$ and repeating the previous steps until we detect a feasible sequence. An overview for deriving the instance-specific bound on the feasible mean is given in Algorithm 2.

The upper bound on the feasible mean can also be used to verify a route's feasibility quickly. If the aggregated mean demand of the customers in a route exceeds the bound, the route is infeasible and can discarded. Exact feasibility verification remains necessary for the routes with an aggregated mean demand less than or equal to $\bar{\mu}$.

Algorithm 2 Instance-specific bound on feasible mean

input: set of customers V' and general upper bound on mean $\bar{\mu}$
while no feasible customer sequence found **do**
  solve model (8) to obtain a value for l
  $C \leftarrow \text{PSSP}(\bar{\mu}, l)$ ▷ obtain all $S \subseteq V'$ with $\sum_{i \in S} \mu_i = \bar{\mu}$ and $|S| = l$
  for $S \in C$ **do**
    assess feasibility under perfect information (Section 4.1)
    if feasible under perfect information **then**
      for every possible sequence of customers in subset S **do**
        compute collective fill rate
        if $\geq F$ **then**
          return $\bar{\mu}$
  $\bar{\mu} \leftarrow \bar{\mu} - 1$ and $l \leftarrow 1$

5 Branch-price-and-cut algorithm

This section proposes a branch-price-and-cut algorithm (BP&C) for the fair stochastic vehicle routing problem with partial deliveries. In this algorithm, column generation and constraint generation (cut separation) techniques are employed to generate new decision variables and constraints dynamically and add these to the linear relaxation of the master problem, hereafter referred to as the restricted master problem (RMP). This way, the number of decision variables and constraints can be significantly reduced. To strengthen the RMP formulation, we propose two sets of valid inequalities (Section 5.1). The algorithm starts by solving the RMP with an initial set of columns (back-and-forward routes to each customer). Next, a pricing algorithm is used to identify the most promising routes to add to the RMP. In Sections 5.2 and 5.3, we introduce the pricing problem and propose three column generation algorithms to search for new routes: exact labeling, heuristic labeling, and tabu search. In Section 5.4, we describe two types of completion bounds that we employ to control the growth of the labels in the labeling algorithms. We iteratively solve the RMP and pricing problem until no new columns with negative reduced costs are generated. Finally, in Section 5.5, we detail the cut separation and branching approaches that we used to reduce the solution space and obtain an optimal integer solution.

5.1 Valid inequalities

To strengthen the linear relaxation of the set-partitioning formulation, we include two sets of cuts: rounded capacity cuts and subset row cuts. Rounded capacity cuts (RCCs) are defined for subsets of customers and enforce lower bounds on the number of routes visiting customer subsets:

$$\sum_{r \in R} \sum_{i \in S} \sum_{j \notin S} a_{ij}^r y_r \geq \left\lceil \frac{\sum_{i \in S} \mu_s}{\bar{\mu}} \right\rceil, \qquad \forall S \subseteq V', \tag{9}$$

where a_{ij}^r is a binary coefficient equal to 1 if route $r \in R$ traverses arc $(i, j) \in A$ and $\bar{\mu}$ is the upper bound on the feasible mean as derived in Section 4.2.

Subset rows cuts (SRCs) proposed by Jepsen et al. (2008) may further help strengthen the formulation as they provide upper bounds on the number of routes visiting customers from subset S, with $S \subseteq V'$. In this work, we consider subset row cuts on triplets of customers:

$$\sum_{r \in R} \left\lfloor \frac{1}{2} \sum_{i \in S} \mathbb{1}(i \in S) \right\rfloor y_r \leq 1, \qquad \forall S \subseteq V', \tag{10}$$

where $\mathbb{1}(i \in S)$ is an indicator function equal to 1 if customer i is in subset S and 0 otherwise.

5.2 Pricing problem

In column generation, new columns are generated by solving a pricing problem. To formally define the pricing problem, let $\mathcal{S}^{\mathrm{RCC}}$ and $\mathcal{S}^{\mathrm{SRC}}$ denote the sets of customer sets for which RCCs and SRCs have been obtained and π_i ($i \in V'$), β_S ($S \in \mathcal{S}^{\mathrm{RCC}}$), and γ_S ($S \in \mathcal{S}^{\mathrm{SRC}}$) be the dual values corresponding to Constraints (5), (9) and (10), respectively. The pricing problem can now be formulated as a variant of the elementary shortest path problem with resource constraints (ESPPRC; Feillet et al., 2004), where the reduced cost of a path is equal to

$$c'_r = c_r - \sum_{i \in C} \pi_i - \sum_{S \in \mathcal{S}^{\mathrm{RCC}}} \sum_{i \in S} \sum_{j \notin S} a^r_{ij} \beta_S - \sum_{S \in \mathcal{S}^{\mathrm{SRC}}} \left\lfloor \frac{1}{2} \sum_{i \in S} a_{ir} \gamma_S \right\rfloor . \tag{11}$$

The objective of the pricing problem is to find a route with a negative reduced cost ($c'_r < 0$). We refer to Appendix 9.1 for a complete arc-based formulation of the pricing problem.

5.3 Column generators

In each iteration of column generation, we search for at most N_C columns with negative reduced costs and add them to the RMP. As there is no need to solve the ESPPRC to optimality in each iteration, we first search for new columns with two heuristic column generators: tabu search and heuristic labeling. If neither heuristic produces any new columns, we proceed with exact labeling. If no columns with negative reduced costs are detected again, we conclude that the current solution to the RMP is optimal. In the remainder of this section, we discuss each algorithm in detail.

Exact labeling.

The ESPPRC is often solved with a labeling algorithm, where each label in the labeling algorithm represents a partial path in graph G. Given the (backward) dynamic programming model for computing the collective fill rate on a sequence of customers (see Equations (1)), we proceed with backward labeling and consider elementary paths only. Table 1 provides an overview of the attributes of a label. Following the approach in Beasley and Christofides (1989) and Feillet et al. (2004), we use a binary vector to keep track of the nodes that can be reached from each label. A node is classified as unreachable if the current label has already visited it, or if visiting the node would result in a collective fill rate below the target fill rate F. However, updating the collective fill rate of a label with every extension to a new node requires a significant amount of computation time and memory. Instead, we assess feasibility concerning the collective fill rate by comparing the aggregated mean demand of the customers visited so far to the upper bound on the feasible mean ($\bar{\mu}$). If the aggregated mean demand does not exceed $\bar{\mu}$, the label is considered feasible (for now). When returning a label would result in a route with a negative reduced cost, we compute its collective fill rate and add the route to the RMP only if its fill rate meets the target fill rate. Otherwise, we discard the label. This way,

Table 1: Label attributes.

Notation	Description
$n \in V$	Last node visited
$c \in \mathbb{R}$	Total cost accumulated
$\phi \in \mathbb{R}$	Sum of dual values
$\mu \in \mathbb{N}$	Aggregated mean demand
$u_i \in \mathbb{B}$	Indicator equal to 1 if node $i \in V'$ cannot be reached from the current label, 0 otherwise.
$\sigma_S \in \mathbb{N}$	Number of customers visited from subset $S \in \mathcal{S}^{\mathrm{SRC}}$

we avoid accepting infeasible routes and reduce the number of collective fill rate evaluations, but at the cost of an increase in the number of labels to consider. However, preliminary experiments showed that this drawback does not outweigh the significant speed up in computation time when compared to computing the collective fill rate for every possible label extension.

We iteratively select the label with the lowest reduced cost ($\mathcal{L}(c) - \mathcal{L}(\phi)$) to be extended. For each customer $i \in V'$, we verify whether an extension of label $\mathcal{L}$ to customer i is feasible. An extension is feasible if the customer is reachable ($\mathcal{L}(u_i) = 0$) and the aggregated mean does not exceed the upper bound on the feasible mean ($\mathcal{L}(\mu) + \mu_i \leq \bar{\mu}$). If feasible, label $\mathcal{L}$ is extended to customer $i \in V'$, resulting in a new label $\mathcal{L}'$:

$$\begin{aligned}
\mathcal{L}'(n) &= i \\
\mathcal{L}'(c) &= \mathcal{L}(c) + c_{\mathcal{L}(n),i} \\
\mathcal{L}'(\phi) &= \mathcal{L}(\phi) + \sum_{S \in \mathcal{S}^{\mathrm{RCC}}} \mathbb{1}(i \in S, \mathcal{L}(n) \notin S)\beta_S + \sum_{S \in \mathcal{S}^{\mathrm{SRC}}} \mathbb{1}(i \in S) \cdot \mathbb{1}(\mathcal{L}'(\sigma_S) = 2)\gamma_S \\
\mathcal{L}'(\mu) &= \mathcal{L}(\mu) + \mu_i \\
\mathcal{L}'(u_i) &= \begin{cases} 1 & \text{if } \mathcal{L}(u_i) = 1 \\ 1 & \text{if } \mathcal{L}(\mu) + \mu_i > \bar{\mu} \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } i \in V' \\
\mathcal{L}'(\sigma_S) &= \begin{cases} \mathcal{L}(\sigma_S) + 1 & \text{if } i \in S \\ \mathcal{L}(\sigma_S + 1) & \text{otherwise} \end{cases} \qquad \text{for } S \in \mathcal{S}^{\mathrm{SRC}}
\end{aligned}$$

The efficiency of the labeling algorithm strongly depends on its ability to prune nonpromising paths. This is often achieved with completion bounds and dominance rules. A completion bound is a lower bound on the reduced cost of all routes that can be created from the current label. In Section 5.4, we elaborate on the two types of completion bounds considered in this work. Dominance rules are imposed on pairs of labels $(\mathcal{L}_1, \mathcal{L}_2)$ and state that label $\mathcal{L}_1$ dominates label $\mathcal{L}_2$ if any extension from label $\mathcal{L}_1$ would result in a path with lower reduced cost and fewer resources used compared to the same extension from $\mathcal{L}_2$. In our setting, one possible resource is the aggregated mean demand. However, since we accept labels that pass the weak constant-time feasibility check ($\mathcal{L}(\mu) \leq \bar{\mu}$), feasible labels could be pruned by infeasible labels, leading to suboptimal results. Therefore, no dominance rules are employed in the exact labeling algorithm.

Heuristic labeling.

To accelerate the search for new columns, we propose a heuristic labeling algorithm in which we relax the fill rate constraint and impose dominance rules. Specifically, we substitute fill rate constraint (3) with a constraint on individual expected fill rates. This enables us to replace the stochastic demand of each customer $i \in V'$ by its minimum allocation required (x_i^-) such that its individual expected fill rate meets the target fill rate:

$$x_i^- = \min\{q : f_i(q) \geq F\}. \tag{12}$$

We use this value to transform the problem into a deterministic CVRP with customer demand x_i^-, $i \in V'$. It is important to note that this approach ignores the possibility of shifting capacity from one customer to another. For instance, if the demand of customer $i \in V'$ is less than the capacity reserved for this customer ($\xi_i < x_i^-$), we could divide the remaining capacity ($x_i^- - \xi_i$) over the customers yet to visit and obtain higher fill rates than we initially computed. Incorporating this possibility will reduce the capacity requirement for any customer subset and result in solutions with lower routing costs. Moreover, it is important to note that even if the individual expected fill rates of the customers in a route satisfy the target fill rate, this may not hold for the route's collective fill rate as the measures are not comparable. Individual fill rates are based on distinct distributions, while collective fill rates are derived from joint demand distributions. Therefore, whenever we create a route with a negative reduced cost, we evaluate its collective fill rate and add it to the RMP only if its collective fill rate meets the target fill rate.

We consider a simple example to demonstrate the differences between the two types of fill rates. Suppose that we have a target fill rate of $F = 0.95$ and two customers with demands following shifted Poisson distributions: $\xi(1) \sim \text{POIS}(\lambda_1 = 5) + 1$ and $\xi(2) \sim \text{POIS}(\lambda_2 = 12) + 1$. To ensure that the expected fill rates of each customer meet the target fill rate, we need to reserve $x_1^- = 6$ and $x_2^- = 14$ units of capacity, which yields fill rates of $f_1(x_1^-) = 0.9511$ and $f_2(x_2^-) = 0.9671$. Their collective fill rate depends on the order in which they are visited and the available capacity. With $Q = x_1^- + x_2^- = 20$, the collective fill rates equal 0.9709 and 0.9661. However, the target fill rate can already be achieved with $Q = 19$ (0.9563 and 0.9508, respectively). The latter shows that taking flexibility into account reduces the capacity needed to achieve the target fill rate.

The approach is heuristic since we discard routes that are infeasible based on the individual fill rates of the customers in the route, even though their collective fill rates may meet the target fill rate. Nonetheless, it will simplify our search for routes that are also feasible when considering individual fill rates only.

The heuristic labeling algorithm uses the same labels as exact labeling, with two small changes. Attribute μ now stores the sum of minimum allocations of each customer (x_i^-) in the path, and attribute m is added to keep track of the number of nodes the label cannot reach. To control the growth of the labels, we propose the following dominance rule:

Definition 1 (Dominance rule)**.** *Label $\mathcal{L}_1$ dominates label $\mathcal{L}_2$ if and only if*

$$\mathcal{L}_1(n) = \mathcal{L}_2(n) \tag{13a}$$

$$\mathcal{L}_1(c) - \mathcal{L}_1(\phi) \leq \mathcal{L}_2(c) - \mathcal{L}_2(\phi) \tag{13b}$$

$$\mathcal{L}_1(\mu) \leq \mathcal{L}_2(\mu) \tag{13c}$$

$$\mathcal{L}_1(m) \leq \mathcal{L}_2(m) \tag{13d}$$

$$\mathcal{L}_1(u_i) \leq \mathcal{L}_2(u_i) \qquad \forall i \in V' \tag{13e}$$

$$\mathcal{L}_1(\sigma_S) \leq \mathcal{L}_2(\sigma_S) \qquad \forall S \in \mathcal{S}^{SRC} \tag{13f}$$

where one of the inequalities (13b)-(13f) *has to hold strictly.*

Definition 1 specifies that if $\mathcal{L}_1$ and $\mathcal{L}_2$ represent two partial paths satisfying Conditions (13a)-(13f), extending label $\mathcal{L}_1$ will always result in a route with lower (or equal) reduced cost. Condition (13a) ensures that both labels have the same last visited node. Condition (13b) requires that the reduced cost of label $\mathcal{L}_1$ is at most equal to the reduced cost of label $\mathcal{L}_2$. Furthermore, Condition (13c) concerns the resource consumption and ensures that label $\mathcal{L}_1$ has at least as many resources left as label $\mathcal{L}_2$. Conditions (13d)-(13e) are related to the unreachable vector and assure that any customer that can be reached from label $\mathcal{L}_2$ can also be reached from label $\mathcal{L}_1$. Finally, Condition (13f) relates to the subset row inequalities.

To reduce the time spent on the dominance checks, we sort the labels in nondecreasing order of reduced cost and group them per last node visited (n) and the number of unreachable nodes (m). Whenever we extend a label $\mathcal{L}$ to a new label L', we process dominance. For each last node visited, we consider all groups with fewer unreachable nodes ($\mathcal{L}''(m) < \mathcal{L}'(m)$) and check whether any of the labels dominate the new label $\mathcal{L}'$. If not, we proceed with checking dominance between label $\mathcal{L}'$ and all labels with the same last node visited and the same number of unreachable nodes ($\mathcal{L}''(m) = \mathcal{L}'(m)$). Finally, if label $\mathcal{L}'$ has not been dominated, we process all labels with the same last visited node and a larger number of unreachable nodes ($\mathcal{L}''(m) > \mathcal{L}'(m)$), and discard all labels that are dominated by label $\mathcal{L}'$.

Tabu search.

Our third column generator is a tabu search algorithm. It takes an initial route as input and iteratively applies neighborhood search operators to obtain new candidate routes (neighbors). The neighbor with the lowest cost is selected as the new route to improve upon, even if its costs are higher than the cost of the current route (Glover and Laguna, 1998). We allow at most $I_{\max}$ iterations per route and prevent cycles by forbidding recent moves for a number of iterations. Similar to Desaulniers et al. (2008), we perform tabu search on the routes in the current solution to the RMP.

We consider two neighborhood search operators. The first neighborhood contains all feasible routes obtained from the current route by removing a single customer. The second neighborhood is constructed with an insertion operator, where each customer $i \in V'$ that is not yet visited is inserted in its cheapest location in the current route. We allow only feasible routes to be included in the pool of neighbors. A route is considered to be feasible if its aggregated mean does not exceed the upper bound on the feasible mean and it does not violate any of the branching decisions made in the current node of the branch and bound tree. If a neighbor has negative reduced cost, we compute its collective fill rate, and add the route to the RMP only if its collective fill rate meets the target fill rate.

5.4 Completion bounds

Completion bounds provide lower bounds on the reduced cost of all routes that can be obtained from a label and will help us identify and prune nonpromising labels. If the completion bound of a label is nonnegative, no route with a negative reduced cost can be generated from the label, and we can safely discard the label. To simplify the derivation of the bounds, we ignore the dual values associated with Constraints (10), since their dual values will be nonpositive and only increase the value of the completion bound. We set the reduced cost of traveling from customer $i \in V'$ to customer $j \in V'$ equal to $\phi_{ij} = \pi_i + \sum_{S \in \mathcal{S}^{\text{RCC}}} \mathbb{1}(i \in S, j \notin S)\beta_S$. In each iteration of column generation, after solving the master problem, we update the bounds with the current dual values of the RMP to enable constant-time retrieval of the bounds during the labeling algorithms.

We consider two sets of completion bounds. The first set is derived by solving a knapsack problem where each customer $i \in V'$ represent an item with value and weight equal to δ_i and μ_i, respectively. The value of the knapsack bound on customer set $S \subseteq V'$ with capacity value $q \in \{0, \dots, \bar{\mu}\}$ is denoted by $KS\big(S, q\big)$. The second set is obtained by solving a resource-constrained shortest path problem with two-cycle (RCSP-2CE) elimination. Let $G' = (V, A)$ be a copy of graph $G = (V, A)$. The cost of arc $(i, j) \in A$ is equal to $c_{ij} - \phi_{ij}$ if $j \in V'$, and c_{ij} otherwise. Visiting node $i \in V'$ consumes μ_i units of capacity. We denote the value of the RCSP-2CE obtained on customer set $S \subseteq V'$ from node $i \in V'$ to the depot with q units of capacity left by $RCSP(S, i, q)$. The completion bound on label $\mathcal{L}$ is now derived as:

$$CB(\mathcal{L}) = \mathcal{L}(c) - \mathcal{L}(\phi) - \max\big\{KS(S, q), RCSP\big(S, \mathcal{L}(n), q\big)\big\} \tag{14}$$

with $S = V'$ and $q = \bar{\mu} - \mathcal{L}(\mu)$.

Stronger bounds may be derived by excluding customers that are already visited. To this end, we also compute completion bounds on subsets of customers. Let set $\mathcal{M}$ contain the m customers with the largest dual value-to-mean ratios (δ_i/μ_i with $\delta_i = \max_{(i,j)\in A} \phi_{ij}$). Each time after solving the master problem, we calculate the completion bounds on every subset of customers $V' \setminus M$ with $M \subseteq \mathcal{M}$. In the labeling algorithms, when deriving a completion bound on label $\mathcal{L}$, set S contains all customers that are reachable from label $\mathcal{L}$, that is $S = V' \setminus (\mathcal{M} \cap \{i \in V' : \mathcal{L}(u_i) = 1\})$. Finally, to reduce the time spent on pre-computing bounds, we only evaluate them for capacity values up to $\rho\bar{\mu}$ with $0 \leq \rho \leq 1$.

5.5 Branch and cut

An initial valid lower bound on the objective value is obtained by iteratively applying column generation until no more columns with negative reduced costs are found. To improve upon this bound, we proceed with cut separation. We iteratively search for any violated RCCs, add them to the master problem, and apply column generation to search for any new columns. If no violated RCCs are detected, we continue with separating SRCs. The RCCs and SRCs are separated with the CVRPSEP package by Lysgaard et al. (2004) and enumeration, respectively. In each iteration, we allow at most $N^{\text{SRC}}_{\text{iter}}$ new SRCs and at most $N^{\text{SRC}}_{\text{cust}}$ new SRCs per customer. We repeat these rounds of column and cut generation until no new columns and cuts are found.

The resulting solution to the RMP may contain fractional variables. To obtain an optimal integer solution to the original problem, we proceed with branch-and-bound. In branch-and-bound, we split ("branch") the problem up into smaller and smaller subproblems ("nodes") and iteratively select one to be solved. Let R' be the set of routes (columns) generated so far. If the total value of routes in the current solution to the RMP ($\bar{y} = \sum_{r \in R'} y_r$) is fractional, we create the following two branches. The first branch restricts the number of routes to be less than or equal to $\bar{y}$ rounded down, whereas the second branch restricts the number of routes to be larger than or equal to $\bar{y}$ rounded up. If the sum of routing variables is integer, we perform branching on the arc with the most fractional value ($\min_{(i,j)\in A} |\sum_{r \in R'} a^r_{ij} y_r - 0.5|$). If multiple arcs qualify, we select the arc that, when removed, leads to the largest increase in the lower bound value when considering the columns and rows generated so far. To select the next node to be solved, we employ the best node first strategy, i.e., we select the node with the lowest lower bound as the next node to be evaluated. From Section

5.3, we know that each SRC requires a separate attribute in the labeling algorithm. Therefore, SRCs are only separated in the root node. The algorithm terminates when all nodes have been explored, a time limit is reached, or the optimality gap drops below 1%.

To speed up the BP&C procedure, we propose two additional techniques. First, to prevent the generation of the same column in different nodes of the branch-and-bound tree, we utilize a general route pool that contains all generated routes. When selecting a node to be solved, we iterate over all routes in the route pool, initialize the corresponding RMP with the routes that comply with the branching decisions of the current node, and proceed with solving it. The second technique relates to searching for new best-known integer solutions. In the algorithm described so far, integer solutions are only detected when a subproblem returns an integer solution. To enable faster detection of promising integer solutions, whenever at least N^C new columns have been added to the RMP, we solve model (4)-(6) with the columns generated so far and update the upper bound accordingly.

6 Alternative models

Instead of solving the model with collective fill rates, one could opt for solving a simpler non-stochastic version of the problem. In this section, we introduce three alternative models. In Section **??**, we compare the results obtained with these alternative models to the results obtained with our proposed model to assess the overall value of our proposed formulation. The main difference between all four models is in their definition of a feasible route. Recall that in our proposed model (model CFR), route $r = (i_0, i_1, \ldots, i_m, i_{m+1})$ with $i_0 = i_{m+1} = 0$ and $i_k \in V'$ for $k \in \{1, \ldots, m\}$ is considered to be feasible if $_r \geq F$.

For the first two alternative models, we set the demand of each customer equal to its expected demand value ($\xi_i = \mu_i$, $i \in V'$). In model D, we require that the sum of demands of the customers visited in a route fits within the vehicle capacity. Hence, route r is feasible if

$$\sum_{k=1}^{m} \mu_{i_k} \leq Q. \tag{15}$$

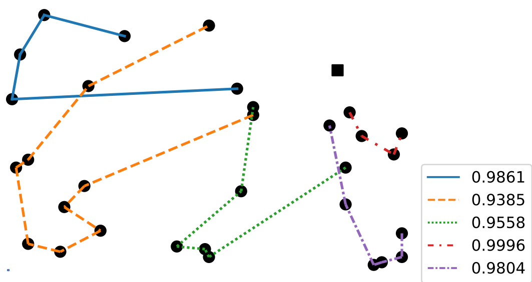
Model DP relaxes the requirement of fully serving the expected demand and allows partial deliveries. Here, route r is feasible if

$$\sum_{k=1}^{m} \mu_{i_k} \leq F \cdot Q. \tag{16}$$

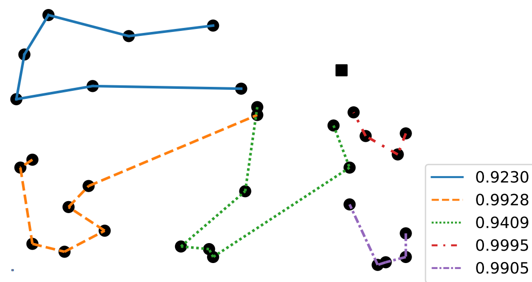
The third alternative was already introduced in Section 5.3 and involves replacing collective fill rate constraint (3) with a constraint on individual expected fill rates. Specifically, we replace the stochastic demand of customer $i \in V'$ with the minimum number of units that should be reserved for this customer (x_i^-) to ensure that its expected fill rate meets the target fill rate (see Model (12)). We refer to this model as the model with individual fill rates (model IFR) and consider route r to be feasible if

$$\sum_{k=1}^{m} x_{i_k}^- \leq Q. \tag{17}$$

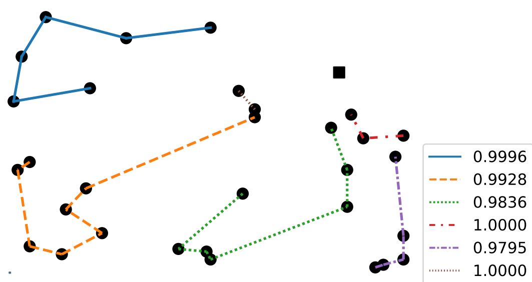
The alternative models serve to assess the overall value of our proposed approach, which explicitly accounts for stochastic demands, partial deliveries, and flexible allocations as a means to reduce overall costs while



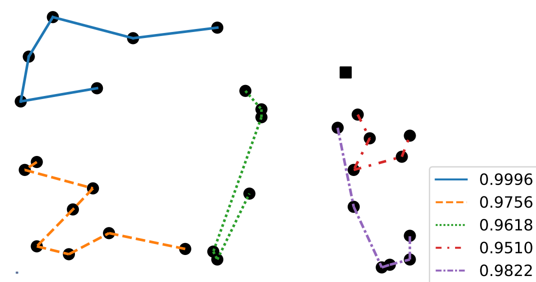
(a) Model D (858)



(b) Model DP (839)



(c) Model IFR (889)



(d) Model CFR (859)

Figure 2: Solutions obtained with four different models, where the objective value is given in brackets and the corresponding collective fill rates are given in the legend. The arcs from starting and ending at the depot are excluded from the plots.

ensuring fair service to all customers.

For each of the models, Figure 2 shows the solution obtained on the same instance. To enhance the graph's clarity, arcs linked to the depot have been excluded from the plots. The objective values and collective fill rates corresponding to each solution are given in brackets and legends, respectively. We observe that the solutions to models D and DP contain at least one route with a collective fill rate below the target fill rate. On the other hand, all routes in the solution to models IFR and CFR are feasible, but at the cost of a much higher objective value in the case of model IFR. By contrast, model CFR has only a 0.11% higher objective value than model D. Hence, model CFR provides feasible solutions at a minimal additional cost compared to model D.

7 Computational experiments

We use benchmark instances from sets A, B, E, and P of the CVRPLIB (Uchoa et al., 2017) for experimental evaluation. We considered all instances with up to 100 customers and a maximum vehicle capacity of 180. This leaves us with 75 instances in total. When transforming the deterministic instances into instances with stochastic customer demands, we assume that we are dealing with *regular* customers that always have demand ($\xi_i \geq 1, i \in V'$). Let D_i denote the deterministic demand of customer $i \in V'$ as provided in the original instance. In our experiments, we adopt a customer-dependent lower bound on the demand at each customer, which is a fraction α of D_i, specifically, $\underline{d}_i = \max(1, \alpha D_i)$ with $0 < \alpha < 1$ and $\underline{d}_i$ rounded to the nearest integer. The remaining portion of the demand follows a Poisson distribution with mean $\lambda_i = D_i - \underline{d}_i$. Note that λ_i could be equal to zero for small values of D_i and large values of α. For example, if $D_i = 2$

and $\alpha = 0.8$, we have $\underline{d}_i = 2$ and $\lambda_i = 0$. The observed demand of customer i depends on the realization of $d_i \sim \text{POIS}(\lambda_i)$, and is expressed as:

$$\xi_i = \underline{d}_i + d_i. \tag{18}$$

The remainder of this section is structured as follows. Section 7.1 discusses the results obtained with our proposed model. In Section 7.2, we compare these results to those obtained with the alternative models. The impact of different levels of known demand (α) and target fill rates (F) on the solution cost and collective fill rates are studied in Sections 7.3 and 7.4. Finally, in Section 7.5, we investigate the contributions of the different algorithmic components. All computational results are performed on a single thread of AMD Rome 7H12 (2.6GHz) processor with 50 GB of available memory and two hours of computation time, excluding the time for computing the upper bound on the feasible mean. Our implementation is available at https://github.com/nsluijk/FSVRPpd.

7.1 CFR model

We performed preliminary testing on ten instances to select the values for the parameters of the BP&C algorithm and report them in Table 2. Table 3 summarizes the results obtained with the proposed branch-price-and-cut algorithm on the selected instances from sets A, B, E, and P with $\alpha = 0.5$ and $F = 0.95$. The first two columns indicate the instance sets and the number of instances considered. The next set of columns provides the number of instances, the average computation time, the average optimality gap, and the number of instances solved to the 1% optimality gap in the root node. The last two columns display the number of instances that could not be solved within the allocated time and memory.

We observe that on 60 instances, the algorithm returns an integer solution within 1% of the lower bound. For 17 out of the 60 instances, it holds that the difference between the lower and upper bound values is strictly less than 1, i.e., they are solved to optimality. Additionally, it holds that ninety percent of them were already solved to 1 % optimality in the root node, i.e., no branch-and-bound is necessary. The remaining fifteen instances could not be solved within the allocated time (6) or memory (9). Figure 3 shows the number of instances that could be solved within x seconds, for each set separately, as well as all sets together, excluding the instances that could not be solved to a 1% optimality gap. We observe that fifty instances were solved within ten minutes (600 seconds). The results clearly show that the method is efficient, since 60 out of 75 instances are solved to a 1% optimality gap. For detailed results per instance, we refer to Appendix 9.2.

7.2 Comparison to alternative models

We solved all four model formulations (D, DP, IFR, and CFR, as described in Section 6) on a subset of ten instances. Models DP, IFR, and CFR are solved with (adaptations to) the proposed BP&C algorithm to a 1% optimality gap. For model D, we took the guaranteed optimal solutions from CVRPLIB. Table 4 reports the objective values obtained for each instance and model combination. The models are arranged in increasing order of objective values, starting with model DP and followed by model D, CFR, and IFR.

Models DP and D neglect the uncertainty associated with customer demands, which results in routes with a high aggregated mean demand-to-vehicle-capacity ratio and, consequently, lower objective values. Model IFR disregards the possibility of transferring reserved capacity from one customer to another, resulting in sub-optimal solutions with higher objective values than those obtained with model CFR. On some instances, model CFR produces lower objective values than model D. One reason is that some of the routes in the

Table 2: Parameter settings.

Parameter	Description	Value
N^C	Minimum number of new columns for solving the integer formulation	100
N_C	Maximum number of columns per iteration	250
N^{SRC}_{iter}	Maximum number of SRCs to add per iteration	40
N^{SRC}_{cust}	Maximum number of SRCs to add per customer per iteration	3
$SRC_{\text{violation}}$	Minimum violation of SRC	0.1
$I_{\max}$	Maximum number of iterations per route in tabu search	10
I_{tabu}	Maximum number of iterations a move is tabu in tabu search	5
$\|\mathcal{M}\|$	Size of set $\mathcal{M}$	5
ρ	Fraction of capacity value for which we compute completion bounds	0.8
N_{SI}	Number of standard deviations for statistical inference	4
N_{scn}	Number of scenarios for statistical inference	100

Table 3: Summary results on selected instances from Sets A, B, E, and P.

		1% optimality gap				Time limit	Memory limit
Set	#	#	Time (s)[a]	Gap(%)[a]	# Root node[b]	#	#
A	27	23	502	0.60	22	1	3
B	23	14	1122	0.51	12	4	5
E	6	4	133	0.65	4	1	1
P	19	19	462	0.43	16	-	-

[a] averaged over all instances solved to 1% optimality gap, [b] number of instances solved to 1% optimality gap in root node

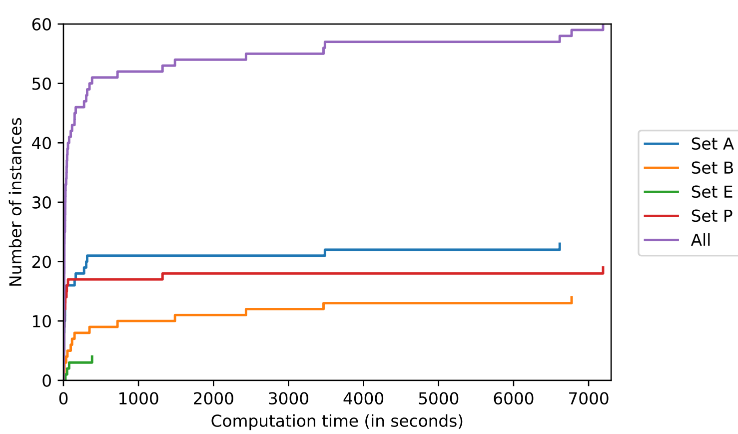


Figure 3: Performance plots on instances solved to 1% optimality gap

Table 4: Objective values obtained with different models

	DP	D	CFR	IFR
A-n39-k6	823	831	834	844
A-n46-k7	911	914	928	971
B-n41-k6	824	829	830	854
B-n45-k6	673	678	690	720
B-n50-k7	740	741	741	750
P-n19-k2	195	212	195	219
P-n20-k2	209	216	209	232
P-n23-k8	525	529	551	568
P-n50-k10	682	696	694	711
P-n51-k10	732	741	746	764

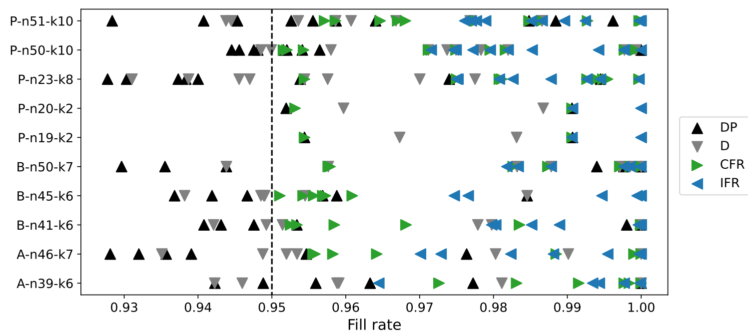


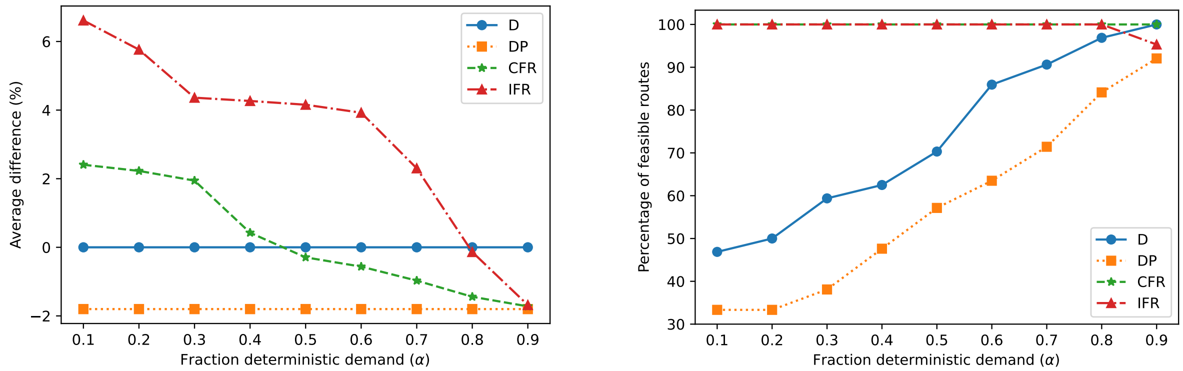
Figure 4: Collective fill rates obtained with four different models.

solutions to model CFR have an aggregated mean demand exceeding the vehicle capacity, making them infeasible in model D. The minimum and maximum difference in objective values with models D and CFR are -8.02% and 4.15%, respectively. On average, we obtain 1.52% higher objective values with model CFR compared to model D, with the benefit of guaranteed feasible routes, as we will show next.

Figure 4 depicts, for each instance and model, the corresponding fill rates of the routes in the solution when evaluated with the demand distributions assumed in model CFR ($\alpha = 0.5$). The collective fill rates depicted for model IFR are well above $F = 0.95$, whereas only some of the fill rates of the routes obtained with models DP and D exceed $F = 0.95$. One explanation for the feasible routes in the solutions to models DP and D is the low aggregated mean demand-to-vehicle-capacity ratio of some of the routes, which indicates that there is sufficient remaining capacity to serve demands that are larger than expected and, consequently, achieve a collective fill rate that meets the target fill rate. In contrast to model CFR, where the collective fill rate requirement is met for all routes, with models D and DP, this fill rate is not met for 29.69% and 42.86% of the routes, respectively. This demonstrates the necessity to account for stochasticity explicitly. Hence, incorporating stochasticity and allowing partial deliveries ensures the solution is feasible at only a marginal cost increase.

7.3 Different levels of uncertainty

In this section, we compare the results obtained with all four models for different levels of uncertainty. To obtain the results, we consider the same subset of instances as in Section 7.2 and solve formulations CFR and IFR for varying values of α, where lower and higher values of α represent cases of high and low uncertainty,



(a) Objective

(b) Feasibility

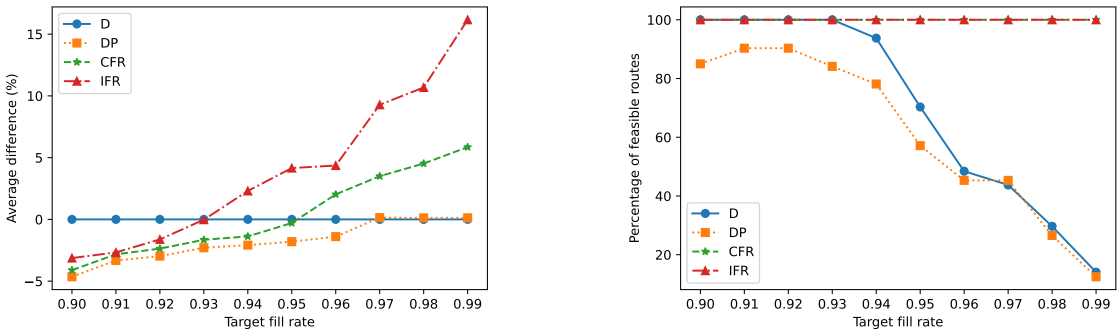
Figure 5: Results obtained with varying levels of uncertainty

respectively. The solutions to formulations D and DP are independent of α and only require re-evaluation of their collective fill rates for each value of α.

Figure 5a displays for $\alpha \in \{0.1, 0.2, \ldots, 0.9\}$ the average difference in objective values between the solutions obtained with model D, and the solutions obtained with model DP, CFR, and IFR, with $F = 0.95$ as before. The difference is constant for models D and DP, as the solutions do not depend on α. The objective values of models CFR and IFR decrease as the uncertainty decreases (α increases). For $\alpha \geq 0.5$, on average, lower objective values are obtained with model CFR compared to model D. Figure 5b shows the percentage of routes with a collective fill rate that meets the target fill rate for each model and value of $\alpha \in \{0.1, 0.2, \ldots, 0.9\}$. For model CFR, this percentage is always equal to 100. For model IFR, this holds for almost all values of α, except for $\alpha = 0.9$, where the percentage drops to 95.15%. Hence, at $\alpha = 0.9$, some solutions to model IFR contain routes with individual fill rates meeting the target fill rate while their collective fill rate is below the target fill rate. We know from Section 3 that this is possible as the two metrics are not comparable. Finally, we observe that as the value for α decreases (uncertainty increases), fewer routes in the solutions to models D and DP meet the target fill rate.

7.4 Different service levels

In the following, we analyze the impact of different service levels (target fill rates) on the corresponding solutions. We consider the same subset of instances as in Section 7.2 and solve models DP, CFR, and IFR for $F \in \{0.90, 0.91, \ldots, 0.99\}$, with $\alpha = 0.5$ as before. Model D requires full demand satisfaction. Hence, its solutions are independent of F. Figure 6a displays, for different values of F, the average difference between the objective values obtained with model D, and models DP, CFR, and IFR. Similar objective values are obtained with models DP, CFR, and IFR at low values for F, but they start to deviate as F increases. Furthermore, as F approaches 1, the difference between the objective values obtained with models D and DP becomes negligible. Figure 6b shows the percentage of routes that meet the target fill rate for each model and $F \in \{0.90, 0.91, \ldots, 0.99\}$. As the value for F increases, the percentage of feasible routes obtained with models D and DP decreases. Overall, we conclude that accounting for stochasticity and partial deliveries (model CFR) leads to solutions that are guaranteed to be feasible concerning the fill rate constraint and have a marginal cost difference compared to model D. For small values of F, this difference is even negative.



(a) Objective

(b) Feasibility

Figure 6: Results obtained with different target fill rates

Table 5: Details results on algorithmic performance.

Setting	Avg. Time(s)	# Root node only
Our proposed BP&C algorithm	10.8	10
without instance-specific bounds on $\bar{\mu}$	57.9	10
without tabu search	15.3	10
without heuristic labeling	14.1	10
without knapsack bounds	11.0	10
without RCSP bounds[1]	533.1	5
without cuts (branch-and-price)	373.3	1
without SRC cuts	11.9	8
without RCC cuts	13.9	10

[1] results reported on the 5 out of 10 instances that were solved to 1 % optimality.

7.5 Contribution of different algorithmic components

Our proposed BP&C algorithm consists of several components that together produce good results. To obtain insights in the contribution of each algorithmic component, we study its impact by solving instances with our proposed BP&C algorithm excluding the selected component. The analysis is performed on ten instances from different instance sets and of varying sizes. Table 5 reports the average computation time and the number of instances solved to 1% optimality in the root node. We observe that the inclusion of instance-specific bounds on $\bar{\mu}$, RCSP bounds, and cuts significantly improve the average computation time and the number of instances solved in the root node. Specifically, only 5 out of 10 instances could be solved to 1 % optimality when the RCSP bounds were excluded from the algorithm. Moreover, by adding cuts, we avoid branch-and-bound since we can solve more instances to a 1% optimality gap in the root node. Finally, we observe that the computation time increases slightly if we exclude either heuristic column generator.

8 Conclusions

A common assumption in the models for the vehicle routing problem with stochastic demands is that all demands must be satisfied. This is achieved by including recourse actions in two-stage stochastic programming formulations or by ensuring with a high probability that all demand fits within the vehicle capacity (chance-constrained formulations). However, recourse actions may not be feasible in practice due to time and/or safety restrictions, e..g, in humanitarian logistics, detours are undesirable due to bad road conditions and/or the high risk of barricades. Additionally, the chance-constrained formulation does not make a distinction between small and large violations of vehicle capacity. One alternative is to relax the assumption of full demand satisfaction and allow partial deliveries to customers. Practical applications of partial deliveries include humanitarian logistics and food rescue programs. To ensure a fair solution for all customers, we required that the expected minimum fill rate over all customers meets the target fill rate. We refer to the resulting problem as the fair stochastic vehicle routing problem with partial deliveries.

In this work, we modeled the problem with a set partitioning formulation and solved it with a branch-price-and-cut algorithm. One advantage of this approach is that the complexity related to the stochastic customer demand can be entirely handled inside the pricing problem. We proposed a method for constructing an upper bound on the feasible mean that can be used as a limiting resource in completion bounds and cuts. To compute the collective fill rate of each route, we employed dynamic programming.

Computational experiments were performed on benchmark instances. When transforming the deterministic instances into instances with stochastic customer demands, we assumed a lower bound on each customer's demand, with the possibility of additional demand that follows a Poisson distribution. Instances with up to 75 customers are solved to a 1% optimality gap. Additionally, we presented an extensive set of numerical experiments in which we compared the results obtained with our proposed model to those obtained with three alternative models with either simplified fill rate derivations and/or deterministic demand. The results showed that with our proposed model, solutions are guaranteed to be feasible at a marginal cost increase only (1.52% on average when compared to the deterministic model).

We conclude by mentioning a few interesting avenues for future research. First, the demand model could be further extended to also incorporate varying lower bound values, where the lower bound on each customer's demand follows some distribution and is revealed at the start of the route, and additional demand may be revealed upon arrival at the customer. Another interesting extension is the inclusion of correlated demands. Also, more research is needed on additional mechanisms to better control the growth of the labels, which will help in solving larger instances. Lastly, while we focused on a welfare-constraining model where fairness was incorporated as a constraint, an alternative approach could be to consider a welfare-optimizing model where the objective function is a convex combination of costs and fairness.

9 Appendix

9.1 Pricing problem

To formally define the pricing problem, let π_i, β_S, and γ_S denote the dual values corresponding to Constraints (5), (9), and (10), respectively. Binary decision variable x_{ij} is equal to 1 if arc $(i,j) \in A$ is used and 0 otherwise. Similarly, binary decision variable z_i equals 1 if customer $i \in V'$ is visited, and 0 otherwise. Finally, integer decision variables u_i indicate the position of customer $i \in V'$ in the route. We now present the arc-based formulation for the pricing problem:

$$\min \sum_{i \in V} \sum_{\substack{j \in V: \\ i \neq j}} c_{ij} x_{ij} - \sum_{i \in C} \pi_i - \sum_{S \in \mathcal{S}^{\mathrm{RCC}}} \sum_{i \in S} \sum_{j \notin S} \beta_S x_{ij} - \sum_{S \in \mathcal{S}^{\mathrm{SRC}}} \left\lfloor \frac{1}{2} \sum_{i \in S} \gamma_S \sum_{j \in V'} x_{ij} \right\rfloor$$

$$
\begin{aligned}
\text{s.t.} \sum_{i \in V'} x_{0j} &= 1, && && (19)\\
\sum_{i \in V'} x_{i0} &= 1, && && (20)\\
\sum_{i \in V \setminus \{j\}} x_{ij} &= z_i, && j \in V', && (21)\\
\sum_{j \in V \setminus \{i\}} x_{ij} &= z_i, && i \in V', && (22)\\
u_j - u_i &\geq 1 - |V'|(1 - x_{ij}), && i, j \in V', i \neq j, && (23)\\
u_i &\leq |V'| z_i, && i \in V' && (24)\\
fr(\mathbf{u}) &\geq F, && && (25)\\
u_0 &= 0, && && (26)\\
u_i &\in \mathbb{N}, && i \in V', && (27)\\
x_{ij} &\in \{0, 1\}, && i, j \in V, i \neq j, && (28)\\
z_i &\in \{0, 1\}, && i \in V'. && (29)
\end{aligned}
$$

The objective is to find the route with minimal reduced cost. Constraints (19) and (20) ensure that one vehicle leaves and enters the depot. Constraints (21) and (22) are the flow conservation constraints and Constraints (23) ensure that the route does not contain any subtours. Constraints (24) require that the position variable u_i can only be positive if customer $i \in V'$ is visited in the route. The position variables are not only included to eliminate subtours, but also to derive the collective fill rate of the constructed route. Let $m = \max_i u_i$ and let $\mathbf{s} \in \mathbb{N}^m$ represent the customer sequence, i.e., $s_k = i$ if $u_i = k$, for all $i \in V'$ and $k \in \{1, \ldots, m\}$. Then, we have

$$fr(\mathbf{u}) = v^{\mathbf{s}}(1, Q). \tag{30}$$

Constraints (25) ensure that the returned route has a collective fill rate that is at least equal to F. Finally, Constraints (26)-(29) set the domains of the variables.

9.2 Detailed computation results

Tables 6 - 9 show the results obtained with the proposed branch-price-and-cut algorithm on the selected instances from sets A, B, E and P with $\alpha = 0.5$ and $F = 0.95$. Columns “LB”, “UB”, “Gap”, “T(s)”, “# Nodes”, “# RCC” and “# SRC” report, respectively, the lower bound, upper bound, optimality gap, solving time in seconds, number of nodes explored in the branch-and-bound tree, number of RCCs added, and, finally, number of SRCs added. A dash in column “T(s)” indicates that the corresponding instance could not be solved with the allocated memory. The remaining columns contain a dash if the algorithm did not produce a valid lower bound for the corresponding instance, which holds for four out of the nine instances with memory issues. On 60 instances, the algorithm returns a solution that is within 1% of the lower bound, and on 17 of them, the difference between the lower and upper bound is less than 1, i.e., they are solved to optimality. The remaining fifteen instances could not be solved within the allocated time (6) or memory (9). Column “# Nodes” shows that most instances were solved in the root node. Finally, we observe that on some instances we already obtain an optimality gap of 1% without adding any cuts.

Table 6: Results on selected instances from set A.

Instance	LB	UB	Gap	T(s)	# Nodes	# RCC	# SRC
A-n32-k5	781.85	784	0.27	5	1	18	0
A-n33-k5	656.43	661	0.70	3	1	0	0
A-n33-k6	737.40	742	0.62	4	1	14	0
A-n34-k5	778.00	778	0.00	7	1	47	23
A-n36-k5	802.81	807	0.52	10	1	40	24
A-n37-k5	669.00	669	0.00	275	1	63	10
A-n37-k6	942.00	949	0.74	12	1	46	50
A-n38-k5	730.70	733	0.31	21	1	49	49
A-n39-k5	824.36	830	0.68	23	1	122	26
A-n39-k6	827.03	834	0.84	9	1	63	20
A-n44-k6	935.67	942	0.68	7	1	26	0
A-n45-k6	953.09	956	0.30	20	3	94	35
A-n45-k7	1141.09	1146	0.43	10	1	100	0
A-n46-k7	921.75	928	0.68	20	1	180	25
A-n48-k7	1081.47	1092	0.97	28	1	113	29
A-n53-k7	997.99	1033	3.51	7200	1	0	0
A-n54-k7	1162.21	1170	0.67	164	1	195	28
A-n55-k9	1072.51	1083	0.98	14	1	121	21
A-n60-k9	1344.65	1358	0.99	3485	1	102	59
A-n61-k9	1029.96	1040	0.97	42	1	125	104
A-n62-k8	-	-	-	-	0	-	-
A-n63-k10	1302.12	1315	0.99	304	1	129	34
A-n63-k9	1618.57	1634	0.95	316	1	344	56
A-n64-k9	1399.09	1416	1.21	-	1	57	83
A-n65-k9	1183.48	1186	0.21	149	1	147	39
A-n69-k9	1161.99	1166	0.35	6615	1	78	136
A-n80-k10	-	-	-	-	0	-	-

Table 7: Results on selected instances from set B.

Instance	LB	UB	Gap	T(s)	# Nodes	# RCC	# SRC
B-n31-k5	672.00	672	0.00	53	1	9	0
B-n34-k5	781.82	789	0.92	97	1	24	0
B-n35-k5	958.00	960	0.21	12	1	16	0
B-n38-k6	764.36	808	5.71	7200	1	31	59
B-n39-k5	555.00	555	0.00	148	1	35	29
B-n41-k6	828.50	830	0.18	12	1	37	0
B-n43-k6	740.58	747	0.87	36	1	79	43
B-n44-k7	872.35	880	0.88	13	1	121	0
B-n45-k5	750.54	761	1.39	-	0	44	0
B-n45-k6	683.58	690	0.94	2433	11	92	61
B-n50-k7	736.39	741	0.63	347	1	41	0
B-n50-k8	1275.20	1300	1.94	7200	3	107	106
B-n51-k7	1035.09	1059	2.31	-	0	29	30
B-n52-k7	746.31	747	0.09	1486	1	131	27
B-n56-k7	-	-	-	-	0	-	-
B-n57-k7	1142.89	1147	0.36	3467	1	102	0
B-n57-k9	1599.89	1610	0.63	6773	3	166	95
B-n63-k10	1489.70	1503	0.89	115	1	134	0
B-n64-k9	852.98	886	3.87	-	0	139	0
B-n66-k9	-	-	-	-	0	-	-
B-n67-k10	1032.44	1038	0.54	720	1	128	55
B-n68-k9	1267.92	1301	2.61	7200	1	305	93
B-n78-k10	1174.14	1278	8.85	7200	0	1	0

Table 8: Results on selected instances from set E.

Instance	LB	UB	Gap	T(s)	# Nodes	# RCC	# SRC
E-n31-k7	365.20	378	3.50	-	440	4160	135
E-n51-k5	521.00	521	0.00	49	1	58	47
E-n76-k10	823.82	832	0.99	76	1	123	105
E-n76-k14	1011.93	1020	0.80	26	1	47	101
E-n76-k8	732.97	739	0.82	382	1	92	150
E-n101-k14	1054.07	1097	4.07	7200	1	72	0

Table 9: Results on selected instances from set P.

Instance	LB	UB	Gap	T(s)	# Nodes	# RCC	# SRC
P-n16-k8	455.50	460	0.99	0	1	9	2
P-n19-k2	195.00	195	0.00	5	1	0	0
P-n20-k2	209.00	209	0.00	7	1	0	0
P-n21-k2	211.00	211	0.00	7	1	0	0
P-n22-k2	216.00	216	0.00	10	1	0	9
P-n23-k8	546.50	551	0.82	0	1	8	0
P-n40-k5	455.69	458	0.51	12	1	22	0
P-n45-k5	510.00	510	0.00	24	1	73	72
P-n50-k10	694.00	694	0.00	6	1	31	57
P-n50-k7	555.00	555	0.00	45	5	76	69
P-n50-k8	622.77	629	1.00	7192	177	72	72
P-n51-k10	742.65	746	0.45	5	1	41	27
P-n55-k10	688.38	692	0.53	1321	3	23	114
P-n55-k15	939.67	945	0.57	2	1	21	0
P-n55-k7	562.80	568	0.92	60	1	35	117
P-n60-k10	738.51	745	0.88	12	1	41	0
P-n60-k15	967.27	969	0.18	6	1	48	41
P-n65-k10	791.52	796	0.57	25	1	39	54
P-n70-k10	821.16	828	0.83	40	1	70	106

References

Agresti, A., Coull, B.A., 1998. Approximate is better than "exact" for interval estimation of binomial proportions. The American Statistician 52, 119–126.

Ak, A., Erera, A.L., 2007. A paired-vehicle recourse strategy for the vehicle-routing problem with stochastic demands. Transportation Science 41, 222–237.

Alkaabneh, F., Shehadeh, K.S., Diabat, A., 2023. Routing and resource allocation in non-profit settings with equity and efficiency measures under demand uncertainty. Transportation Research Part C: Emerging Technologies 149, 104023.

Anaya-Arenas, A.M., Ruiz, A., Renaud, J., 2018. Importance of fairness in humanitarian relief distribution. Production Planning & Control 29, 1145–1157.

Anuar, W.K., Lee, L.S., Pickl, S., Seow, H.V., 2021. Vehicle routing optimisation in humanitarian operations: A survey on modelling and optimisation approaches. Applied Sciences 11, 667.

Balcik, B., Iravani, S., Smilowitz, K., 2014. Multi-vehicle sequential resource allocation for a nonprofit distribution system. IIE Transactions 46, 1279–1297.

Beasley, J.E., Christofides, N., 1989. An algorithm for the resource constrained shortest path problem. Networks 19, 379–394.

Chen, V.X., Hooker, J., 2022. A guide to formulating fairness in an optimization model. Annals of Operations Research .

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

Desaulniers, G., Desrosiers, J., Solomon, M.M., 2006. Column generation. volume 5. Springer Science & Business Media.

Desaulniers, G., Lessard, F., Hadjar, A., 2008. Tabu search, partial elementarity, and generalized k-path inequalities for the vehicle routing problem with time windows. Transportation Science 42, 387–404.

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.

Eisenhandler, O., Tzur, M., 2019. The humanitarian pickup and distribution problem. Operations Research 67, 10–32.

European Commission, 2022. Directorate general for European civil protection and humanitarian aid operations thematic policy document - humanitarian logistics policy. URL: https://data.europa.eu/doi/10.2795/009117, doi:10.2795/009117. Accessed: 2023-03-28.

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., Feillet, D., Poggi, M., Vidal, T., 2022a. Vehicle routing with stochastic demands and partial reoptimization. Transportation Science 56, 1393–1408.

Florio, A.M., Gendreau, M., Hartl, R.F., Minner, S., Vidal, T., 2022b. Recent advances in vehicle routing with stochastic demands: Bayesian learning for correlated demands and elementary branch-price-and-cut. European Journal of Operational Research. doi:10.1016/j.ejor.2022.10.045. ahead of print.

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.

Gansterer, M., Hartl, R.F., 2018. Collaborative vehicle routing: A survey. European Journal of Operational Research 268, 1–12.

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

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

Glover, F., Laguna, M., 1998. Tabu search, in: Handbook of combinatorial optimization. Springer, pp. 2093–2229.

Goodson, J.C., Ohlmann, J.W., Thomas, B.W., 2013. Rollout policies for dynamic solutions to the multivehicle routing problem with stochastic demand and duration limits. Operations Research 61, 138–154.

Gralla, E., Goentzel, J., Fine, C., 2014. Assessing trade-offs among multiple objectives for humanitarian aid delivery using expert preferences. Production and Operations Management 23, 978–989.

Gutjahr, W.J., Nolz, P.C., 2016. Multicriteria optimization in humanitarian aid. European Journal of Operational Research 252, 351–366.

Hoogendoorn, Y., Spliet, R., 2023. An improved integer l-shaped method for the vehicle routing problem with stochastic demands. INFORMS Journal on Computing. doi:10.1287/ijoc.2023.1271. ahead of Print.

Hooker, J.N., 2023. in: Operational Research: Methods and Applications. chapter Ethics and fairness, pp. 98–101. URL: https://arxiv.org/abs/2303.14217.

Huang, M., Smilowitz, K., Balcik, B., 2012. Models for relief routing: Equity, efficiency and efficacy. Transportation Research Part E: Logistics and Transportation Review 48, 2–18.

Ibarra-Rojas, O., Silva-Soto, Y., 2021. Vehicle routing problem considering equity of demand satisfaction. Optimization Letters 15, 2275–2297.

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

Jepsen, M., Petersen, B., Spoorendonk, S., Pisinger, D., 2008. Subset-row inequalities applied to the vehicle-routing problem with time windows. Operations Research 56, 497–511.

Karsu, Ö., Morton, A., 2015. Inequity averse optimization in operational research. European Journal of Operational Research 245, 343–359.

Khorsi, M., Chaharsooghi, S.K., Bozorgi-Amiri, A., Kashan, A.H., 2020. A multi-objective multi-period model for humanitarian relief logistics with split delivery and multiple uses of vehicles. Journal of Systems Science and Systems Engineering 29, 360–378.

Kyriakidis, E.G., Dimitrakos, T.D., 2017. Single vehicle routing problem with a predefined customer sequence, stochastic demands and partial satisfaction of demands, in: Operations Research Proceedings 2015. Springer, pp. 157–164.

Lien, R.W., Iravani, S.M., Smilowitz, K.R., 2014. Sequential resource allocation for nonprofit operations. Operations Research 62, 301–317.

Lu, Y., Yang, C., Yang, J., 2022. A multi-objective humanitarian pickup and delivery vehicle routing problem with drones. Annals of Operations Research , 1–63.

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

Marsh, M.T., Schilling, D.A., 1994. Equity measurement in facility location analysis: A review and framework. European Journal of Operational Research 74, 1–17.

Matl, P., Hartl, R.F., Vidal, T., 2018. Workload equity in vehicle routing problems: A survey and analysis. Transportation Science 52, 239–260.

Nair, D.J., Rey, D., Dixit, V.V., 2017. Fair allocation and cost-effective routing models for food rescue and redistribution. IISE Transactions 49, 1172–1188.

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

Novoa, C., Storer, R., 2009. An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. European Journal of Operational Research 196, 509–515.

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.

Rivera, A.F., Smith, N.R., Ruiz, A., 2023. A systematic literature review of food banks' supply chain operations with a focus on optimization models. Journal of Humanitarian Logistics and Supply Chain Management 13, 10–25.

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 hybrid recourse policy for the vehicle routing problem with stochastic demands. EURO Journal on Transportation and Logistics 8, 269–298.

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

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

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

Sluijk, N., Florio, A.M., Kinable, J., Dellaert, N., Van Woensel, T., 2023. A chance-constrained two-echelon vehicle routing problem with stochastic demands. Transportation Science. 57, 252–272.

Uchoa, E., Pecin, D., Pessoa, A., Poggi, M., Vidal, T., Subramanian, A., 2017. New benchmark instances for the capacitated vehicle routing problem. European Journal of Operational Research 257, 845–858.

Vidal, T., Laporte, G., Matl, P., 2020. A concise guide to existing and emerging vehicle routing problem variants. European Journal of Operational Research 286, 401–416.