Branch-and-Price for Routing with Probabilistic Customers

Felipe Lagos[1], Mathias A. Klapp[2], Alejandro Toriello[1]

[1]H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332

[2]Engineering School
Pontificia Universidad Católica de Chile
Santiago, Chile

felipe.lagos@gatech.edu, maklapp@ing.puc.cl, atoriello@isye.gatech.edu

March 28, 2019

Abstract

We study the Vehicle Routing Problem with Probabilistic Customers (VRP-PC), a two-stage stochastic optimization problem that is a fundamental building block within the broad family of stochastic routing models. In the first stage, a dispatcher determines a set of vehicle routes serving all potential customer locations, before actual requests for service realize. In the second stage, vehicles are dispatched after observing the subset of customers requiring service; a customer not requiring service is skipped from its planned route at execution. The objective is to minimize the expected vehicle travel cost assuming known customer realization probabilities.

We propose a column generation framework to solve the VRP-PC to a given optimality tolerance. Specifically, we present two novel algorithms, one that under-approximates a solution's expected cost, and another that uses its exact expected cost. Each algorithm is equipped with a route pricing mechanism that iteratively improves the approximation precision of a route's reduced cost; this produces fast route insertions at the start of the algorithm and reaches termination conditions at the end of the execution. Compared to branch-and-cut algorithms for the VRP-PC using arc-based formulations, our framework can more readily incorporate sequence-dependent constraints. We provide *a priori* and *a posteriori* performance guarantees for these algorithms, and demonstrate their effectiveness via a computational study on instances with realization probabilities ranging from 0.5 to 0.9.

Keywords: vehicle routing; probabilistic routing; column generation

1 Introduction

The Vehicle Routing Problem (VRP) and its variants are widely studied within the transportation and operations research communities, and used in a variety of applications in freight and urban transportation and logistics, as well as in other areas [17, 27, 49]. Stochastic VRPs are extensions

of their deterministic counterparts where some instance parameters are unknown while planning and/or executing a solution, and decisions must account for this uncertainty. *A priori* VRPs are particular versions of stochastic VRPs modeled as two-stage stochastic optimization problems; see [13] for a survey. In the first stage, some of the parameters are random variables and the decision maker plans an initial solution. In the second stage, the planned solution is executed after the realization of the random parameters. Typically, a simple recourse rule is used to modify the initial plan during its execution. A desired feature in the first stage solution is to proactively account for parameter uncertainty and the second-stage recourse.

We introduce a set partition model to study the *a priori* VRP with probabilistic customers (VRP-PC). In the VRP-PC, the subset of customers requiring service is random and follows a known probability model. In the first stage, the decision maker plans a set of vehicle routes dispatched from the depot and visiting all potential customer locations. Vehicles are dispatched in the second stage after observing the subset of customers requiring service. The second-stage recourse rule modifies first-stage routes by skipping locations without a service requirement, maintaining the established sequence of each route for the visited customers. The objective is to minimize the expected vehicle travel cost, accounting for this recourse rule. The VRP-PC extends the Probabilistic Traveling Salesman Problem P-TSP [31, 32], a single-vehicle version of the problem. Our formulation is the first route-based model for the VRP-PC, and contrasts with the arc-based VRP-PC formulation originally proposed in [25].

1.1 Applications of VRP-PC

There are many applications of the VRP-PC. As an extension of the P-TSP, it may be used in multi-vehicle and constrained problems related to the P-TSP. It is a natural model when the set of customers requiring service is unknown in the planning stage, but it is infeasible or impractical to optimize routes at execution, *e.g.*, [9]; this might happen when the decision maker does not have enough computational resources to optimize routes in real time [3], or when maintaining the initial visit sequence is desirable or required, *i.e.*, because of gains in operational efficiency produced by drivers executing the same plan every day, or because customers expect to meet the same driver at roughly the same time when they require service.

Some application examples include technician routing and scheduling, and cash-in-transit vehicle routing problems (CTVRP). In the technician routing and scheduling problem there is a limited crew of technicians serving requests requiring specialized tools and skills; such a problem arises in public services, telecommunication services and equipment maintenance operations. In such settings, routes that maintain visit sequences make it easier to meet technical requirements; see *e.g.*, [16, 29, 43]. In the CTVRP [45], vehicles are assumed to transport banknotes, coins and other items of high value. Vehicle routes are therefore vulnerable to robberies, and a carefully planned route is desirable to reduce security risks. A priori route planning maintains a daily routine and improves aspects of customer security, such as time window constraints and visit order.

A novel application involves using the VRP-PC as a building block to solve more complex dynamic and stochastic VRPs; see *e.g.*, [4, 40, 52]. In many dynamic settings, the model can be leveraged to design heuristic dynamic policies by maintaining and periodically updating a feasible route plan via a rollout procedure [6, 7, 28, 50]. For example, our particular motivation for studying the VRP-PC stems from urban distribution problems in same-day delivery [35, 36]. In such a setting, customer delivery requests realize dynamically over the operating period as other previously known orders are being prepared, loaded into vehicles and dispatched from a depot to customer locations in vehicle routes. Therefore, an effective dispatch policy requires adapting and reacting to newly revealed information and involves online re-optimization of planned routing decisions. One type of high-quality heuristic policy carries a plan serving a subset of known and pending delivery requests, along with a set of potential customer locations (*e.g.*, neighborhoods, city blocks) to account for future expected routing costs. At the time of dispatch, each route in fact only visits a location if an order realizes there before the vehicle route begins. An *a priori* solution of this kind is not by itself necessarily desirable, as it could result in significant wasted time for the vehicle if executed when many orders do not realize. Nevertheless, one can "roll out" such an *a priori* solution, enforcing in the model that such a route only includes already realized orders; the remainder of the plan is a proxy for the expected duration, length and/or cost of subsequent "average" routes in an "average" day.

In some circumstances, same-day delivery systems may dynamically choose whether to accept customer requests or not. Our model can be used to guide order acceptance decisions while proactively considering potential future orders [34].

1.2 Contribution

We introduce a set covering model for the VRP-PC and propose a novel exact approach for it, which is compatible with a wide variety of route-dependent constraints, including useful sequence-dependent constraints, such as time windows and precedence constraints. An exact algorithm for the P-TSP is proposed in [39], which is a specialized implementation of the integer L-shape method [37]. This method formulates the P-TSP as a two-stage integer linear program with edge-based variables, and replaces the expected route cost in the objective by a variable $\theta \geq 0$. As integer routes are found, the real expected cost of these solutions is evaluated to dynamically generate "optimality cuts" on θ and correct its value.

The integer L-shape method operates on a formulation with edge variables. However, many practical sequence-dependent constraints are difficult to model with edge variables, and often have weak relaxations. In same-day delivery problems, pertinent examples of such constraints include delivery deadlines and order release times.

Therefore, to our knowledge the work on routing with probabilistic customers considering hard sequence-dependent constraints has been restricted to heuristics; see [14, 51]. Nonetheless, in deterministic problems, route-based formulations solved via column generation and branch-and-price (B&P) are arguably more effective to optimize routing models with such constraints (see *e.g.*, [18]). We address this gap in the literature and propose a B&P framework to solve the VRP-PC. In particular, our contributions are:

1. We present two different and independent column generation algorithms, one that underestimates a feasible solution's cost and another that uses its exact expected cost. Both algorithms use route generation subroutines that compute estimates of a route's reduced cost. The algorithms' pricing problems iteratively increase the precision of a route's expected cost estimate, yielding fast route generation at the start while still reaching termination conditions at the end of the execution.

2. We provide *a priori* and *a posteriori* performance guarantees for these algorithms, which allow us to determine solution quality before and after a lower bound on the optimal value is

computed.

3. We implement a prototypical branch-and-price scheme for the VRP-PC and conduct a computational study, concluding that both algorithms find good solutions using only a few number of precision updates. The algorithm's empirical convergence takes few iterations, and depends on the customer realization probabilities.

After closing this section with a literature review, the remainder of the article is organized as follows. Section 2 reviews the B&P approach for the VRP and formulates the VRP-PC. In Section 3 we present two column generation algorithms for the VRP-PC, discuss incorporating them into B&P, and give convergence guarantees and approximate optimality conditions. Section 4 presents a computational study on modified Solomon instances [44], and Section 5 concludes.

1.3 Literature Review

A priori optimization is routinely applied in stochastic routing problems [11, 33, 48]. Different sets of uncertain parameters within VRPs are modeled through different *a priori* optimization problems. For example, the VRP with stochastic travel times (VRP-STT) refers to uncertainty in travel times between locations [38, 41] and the VRP with stochastic demand (VRP-SD) [8, 9] refers to a VRP where the customer demand is a random variable realized at the moment of service.

The *a priori* VRP with probabilistic customers (VRP-PC) is a stochastic VRP where the subset of customers requiring service is random and follows a known probability distribution. The seminal work in [31, 32] formally introduces the Probabilistic Traveling Salesman (P-TSP) and develops a closed-form expression for a route's expected cost in the case of homogeneous probabilities. [5] extends the P-TSP to heterogeneous probability distributions, and additional results include [10, 9, 11, 33].

In [37], the authors present the integer L-shape method to exactly solve *a priori* optimization problems; the algorithm gradually improves an estimate of the expected cost via dynamic generation of cutting planes in an integer program. In [39], the integer L-shape method is used to solve an edge-based formulation of the P-TSP. This method is also implemented in [25] to design an exact algorithm for a VRP with both probabilistic customers and stochastic demand.

Truncation approaches have been proposed by [26] for the P-TSP; they underestimate the

true cost by computing only some terms of the expected cost of a route. In [46], the authors underestimate the true cost using an approximation function; this function balances the amount of speedup for solving algorithms and the quality of the approximation. Such approximation functions have been used to speed up heuristics [12].

For deterministic routing, B&P approaches have been successfully applied to solve many VRP variants to optimality, and readily handle sequence dependent constraints. These approaches formulate the VRP as a set covering route-based formulation; see *e.g.*, [2, 19, 20] and references therein. Recently, B&P has been adapted to solve the VRP-SD [15, 21, 24] and the VRP-STT [23, 47]. We are not aware of a B&P scheme for the VRP-PC, and the technical hurdles needed to overcome probabilistic customers in B&P are in several respects different from these previous works.

The route pricing problem within a VRP set partitioning formulation is the Elementary Shortest Path Problem with Resource Constraints (ESPPRC). Because of the ESPPRC's theoretical complexity and empirical difficulty, a relaxed version called the Shortest Path Problem with Resource Constraints (SPPRC) [18] is usually solved instead; the SPPRC relaxes the elementary condition to allow visiting a customer more than once. The SPPRC with k-cycle elimination [30] is an intermediate relaxation that only allows paths with cycles having at least $k+1$ nodes.

2 Model Formulation and Preliminaries

The deterministic VRP is a starting point to study the more complex probabilistic variant. We start by briefly describing a deterministic VRP set partitioning model and its classic column generation framework, including a relaxation technique for its route pricing subproblem; these concepts are widely studied in the VRP and column generation literature, *e.g.*, [18, 49]. Later, we introduce the VRP with probabilistic customers and discuss how to fit the previous column generation framework in this two-stage stochastic problem. We also show how chance constraints, a useful modeling tool in stochastic optimization, can be incorporated in the model.

2.1 Deterministic VRP

The deterministic VRP entails designing a set of vehicle routes, each starting and ending at a depot, that feasibly serve a set $C = \{1, ..., n\}$ of customers at minimum total travel cost. Each customer $i \in C$ has demand $d_i > 0$ that must be served by one route; that is, we do not allow split deliveries. We consider homogeneous vehicles with capacity $q \geq \max_i d_i$, so that the total demand served on any route does not exceed q. A vehicle can only perform one route, and we may have a limit on the total number of routes. We also assume no fixed route setup costs, though these may be included without substantially changing the model.

Define the set $N := C \cup \{0, n+1\}$ of nodes having all customers plus 0 and $n+1$, both of which represent the depot. A vehicle route departs node 0, visits a subset of customer nodes, and ends at node $n+1$. Traveling between any two nodes $i, j \in N$ costs $c_{ij} \geq 0$ and takes $t_{ij} \geq 0$ time, and we assume both parameter sets satisfy the triangle inequality. When $i \in C$, the value t_{ij} may include service time at i, and we may also have a route duration limit T, with $T \geq t_{ij}$ for all arcs. For the sake of exposition, we describe only vehicle capacity and route duration as limiting resources, but other complex and sequence-dependent constraints, such as service time-windows, order release dates and customer precedence constraints amenable to column generation and B&P may be handled similarly. We only require that all applicable constraint parameters are integers, possibly by re-scaling.

2.2 Column Generation and Branch-and-Price

To specify the column generation framework, define R_n as the set of all feasible vehicles routes; each route $r \in R_n$ corresponds to an elementary path of nodes in C starting at 0, ending at $n+1$ and satisfying any required constraint. Let c_r be the cost of route r and $\alpha_r^i \in \{0, 1\}$ be the number of times $i \in C$ is visited by r.

Define the binary variable y_r, equal to 1 if route r is selected and 0 otherwise. The VRP can then be formulated as a set partitioning integer linear problem, where its linear relaxation is given by

$$\min_{y \geq 0} \sum_{r \in R_n} c_r y_r \tag{1a}$$

$$\text{s.t.} \sum_{r \in R_n} \alpha_r^i y_r = 1, \qquad i \in C. \tag{1b}$$

The number of routes in R_n can be exponentially large as a function of n, making it difficult to solve (1) explicitly with an LP solver. Instead, routes can be generated dynamically using column generation. To solve the VRP set partition LP relaxation, we use an algorithm with two main components. Initially, we solve model (1) only considering a small subset $\tilde{R} \subset R_n$ of feasible routes. The optimal dual solution of this restricted LP is then used to identify profitable columns in $R_n \setminus \tilde{R}$ via a pricing subproblem. We add these new columns to $\tilde{R}$, then execute a new run of the LP solver. The procedure is repeated until no more profitable columns are found in $R_n \setminus \tilde{R}$.

Let ρ_i for $i \in C$ be the dual LP variable related to constraint (1b). The pricing problem to generate routes based on the current restricted LP solution is

$$\min_{r \in R_n} \left\{ c_r - \sum_{i \in C} \alpha_r^i \rho_i \right\}. \tag{2}$$

A non-negative optimal value of (2) certifies optimality for the restricted version of (1). Conversely, a negative value indicates the existence of a profitable route $r \in R_n \setminus \tilde{R}$.

Problem (2) is known as the Elementary Shortest Path Problem with Resource Constraints (ESPPRC), and it is strongly NP-hard [22]. The Shortest Path Problem with Resource Constraints (SPPRC) is a relaxation of (2) that allows multiple visits to a customer; this relaxed version can be solved by dynamic programming (DP) in polynomial time as a function of n and the resources' parameters. In the case of vehicle capacity, the applicable parameter is vehicle capacity q; similarly, for time-based constraints the corresponding parameter is the duration limit T. A relaxed route r can have cycles, but cannot make more than n customer visits, and may in fact be further constrained, since resources are increasingly consumed along any path. We consider k-cycle elimination for the SPPRC (SPPRC-k) [30] to improve relaxation quality; in this setting any path with cycles having fewer than $k+1$ nodes is not allowed. Let R_k be the set of relaxed routes with k-cycle elimination, and observe that since elementary paths exclude all cycles up to length n, this definition is consistent with our use of R_n.

The SPPRC is defined over a network of partial route states $G = (V, A)$. Each state $v \in V$ is specified by a tuple $v = (S_v, d_v, t_v)$, where S_v is a sequence of visited locations in the partial route,

d_v is the vehicle capacity already used by the partial route, and t_v is the partial route's end time. When the problem has other route resources, these are tracked in a similar fashion. The ESPPRC carries the complete node visiting sequence from the depot in S_v, while the SPPRC-k relaxation only records the latest k visited nodes in the sequence. The transition cost between states v and u is defined by $c_{\ell(S_v),\ell(S_u)} - \rho_{\ell(S_v)}$, where $\ell(S)$ is the last node in sequence S.

The DP optimality equations for the SPPRC-k are

$$F([0], 0, 0) = 0, \tag{3a}$$

$$F(S, d, t) = \min_{(i,j)\in N^2} \{F(\bar{S}, \bar{d}, \bar{t}) + (c_{ij} - \rho_i) : (i, j) = (\ell(\bar{S}), \ell(S)), i \neq n + 1, f_{ij}(\bar{d}, \bar{t}) \leq (d, t)\}, \tag{3b}$$

where $\rho_0 = 0$. The value of $F(S, d, t)$ is the smallest possible reduced cost for a partial route starting from the depot, ending with sequence S (where $|S| \leq k$), having consumed resources (d, t). The relaxed route with minimum reduced cost is given by the minimum over all values $F(S, q, T)$ with $\ell(S) = n + 1$. In the recursion, $\bar{S}$ is a sequence with $|\bar{S}| \leq k$ that can be extended to S; that is, either $|\bar{S}| < k$ and appending $\ell(S) = j$ to it yields S, or $|\bar{S}| = k$ and S consists of deleting the first element and appending j. The function f_{ij} is called a resource extension function [18] and models resource consumption, *i.e.*,

$$f_{ij}(d, t) = (d + d_j, t + t_{ij}).$$

The function can be defined more generally and include any other resource consumed along a route. We consider the Bellman-Ford labelling algorithm to solve (3); the number of states satisfies $|V| = O(n^k qT)$, and therefore the running time of the algorithm is $O(n^{k+1} qT)$.

The optimal values of (1) may be fractional; however, branching on the y variables is generally impossible (as the pricing problem cannot be readily updated) and would lead to extremely unbalanced search trees. Instead, for VRP the typical B&P scheme [18] branches on the implied arc variables x_{ij} that indicate whether any route in the solution travels directly from i to j. Adjusting the pricing SPPRC-k model for these branching decisions simply involves forcing or forbidding some actions at certain states. In our implementation, we also initially branch on the number of routes in the solution, as originally proposed in [19].

2.3 VRP with Probabilistic Customers

We now consider a VRP with probabilistic customers (VRP-PC), where an initial solution covering all customers C is planned, but only a subset of customers will actually require service. As a recourse rule, a customer that does not require service is skipped in its corresponding route, while keeping the rest of the route's sequence; for many VRP constraints of interest (such as vehicle capacity and route duration), this ensures that if the planned route is feasible when all customers are serviced, it remains feasible for any realized subset of customers in the probabilistic context. We assume a known, independent customer realization probability $p_i \in (0, 1]$ for each $i \in C$ and use $p_0 = p_{n+1} = 1$. The objective is to minimize the expected vehicle travel cost. The problem's stochasticity only affects expected route costs, and therefore any constraints are managed identically to the deterministic VRP.

For a route $r \in R_n$, let $n_r \leq n$ be the number of planned customer visits, and let $r(i) \in N$ represent the i-th planned visit in r (with $r(0) = 0$, $r(n_r + 1) = n + 1$). The expected cost $\mathbb{E}(c_r)$ of the route is

$$\begin{aligned}\mathbb{E}(c_r) &= \sum_{i=0}^{n_r} \sum_{j=i+1}^{n_r+1} \left(p_{r(i)} p_{r(j)} \prod_{\ell=i+1}^{j-1} (1 - p_{r(\ell)}) \right) c_{r(i),r(j)} \\ &= \sum_{\ell=1}^{n_r} \sum_{i=0}^{n_r-\ell+1} \left(p_{r(i)} p_{r(i+\ell)} \prod_{j=i+1}^{i+\ell-1} (1 - p_{r(j)}) \right) c_{r(i),r(i+\ell)} =: \sum_{\ell=1}^{n_r} H_r^\ell. \end{aligned} \tag{4}$$

This expected cost can be computed as a sum of n_r nonnegative terms as in (4), where each H_r^ℓ includes the expected costs corresponding to arcs that skip exactly $\ell - 1$ customers. For example, the term $H_r^1 = \sum_{i \leq n_r} p_{r(i)} p_{r(i+1)} c_{r(i),r(i+1)}$ considers all arcs between consecutive customers. The term H_r^2 includes all arcs that skip one customer in the sequence, and so on. Since they are all non-negative, a summation of any subset of the H_r^ℓ terms provides a lower bound for $\mathbb{E}(c_r)$. In particular, this includes the possible case in which all the customers in a route do no request service, $H_r^{n_r+1}$, and the cost of that realization is zero, $H_r^{n_r+1} = 0$.

A set partitioning relaxation for the VRP-PC is

$$f(R_k) := \min_{y \geq 0} \left\{ \sum_{r \in R_k} \mathbb{E}(c_r) y_r : \sum_{r \in R_k} \alpha_r^i y_r = 1, i \in C \right\}. \tag{5}$$

Compared to a deterministic VRP relaxation, the feasible region remains unaltered, but the objective function considers each route's expected cost. As in the deterministic case, we can relax the set of feasible routes from R_n to a larger set R_k that only excludes k-cycles; we make the dependence on the set of feasible routes explicit and use $f(R_k)$ to denote the optimal value of this relaxation as a function of the considered route set. In this case, the definition of $\mathbb{E}(c_r)$ can be extended to include routes with repeated customer visits, by simply setting $c_{r(i),r(j)} = 0$ when $r(i) = r(j)$ and keeping all other values the same. In other words, this definition treats repeated visits to customer i in a route as independent copies of it, each of which requires service independently with probability p_i. As in the deterministic problem, such a definition guarantees that routes with repeated visits never appear in an optimal integer solution.

2.4 Chance Constraints

As stated, our model handles resource consumption deterministically. For example, we construct routes that satisfy vehicle capacity even if every customer realizes. In our motivating applications [35, 36] this is indeed necessary. Furthermore, if realization probabilities are high, say $p_i \geq 0.9$ for all customers i, it may be desirable to guarantee the route's feasibility under any circumstance. Therefore, most of our exposition and the instances in our computational study use this approach.

However, in other cases, guaranteeing route feasibility with absolute certainty may result in overly conservative routes with significant amounts of wasted capacity. A systematic approach to this issue involves replacing a deterministic constraint with a chance constraint, which for example stipulates that the probability of realized customers' demand exceeding capacity should be small. By incorporating a chance constraint, we keep the same two-stage planning method used in the deterministic case, but we allow the planned routes to possibly violate resource capacity, as long as the violation is unlikely. This technique has been explored before for the VRP-SD [21]; we suggest one possible way to include it here.

Let D_i be a random variable representing the realized demand at customer i; so D_i follows a scaled Bernoulli distribution with $\mathbb{P}(D_i = d_i) = p_i$ and $\mathbb{P}(D_i = 0) = 1 - p_i$. For a planned route r, we could replace the deterministic vehicle capacity constraint $\sum_{i \leq n_r} d_{r(i)} \leq q$ with the chance

constraint

$$\mathbb{P}\left(\sum_{i=1}^{n_r} D_{r(i)} > q\right) \leq \eta,$$

for an appropriately chosen tolerance $\eta > 0$; for example, $\eta = 0.1$ means the vehicle's capacity can satisfy realized demand with at least 90% probability.

The issue is how to model such a chance constraint so that it is amenable to pricing via a recursion similar to (3). Applying a Chernoff bound, we obtain

$$\mathbb{P}\left(\sum_{i=1}^{n_r} D_{r(i)} > q\right) \leq e^{-q\tau} \prod_{i=1}^{n_r} \mathbb{E}\big[e^{D_{r(i)}\tau}\big] = e^{-q\tau} \prod_{i=1}^{n_r} \big(1 - p_{r(i)} + p_{r(i)} e^{d_{r(i)}\tau}\big), \quad \forall\, \tau > 0,$$

where we use the fact that the D_i's are independent, and $\tau > 0$ can be chosen based on problem parameters. If the quantity on the right does not exceed the tolerance η, we guarantee that the route is feasible for the chance constraint. In other words, the route is feasible if for some $\tau > 0$ we have

$$e^{-q\tau} \prod_{i=1}^{n_r} \big(1 - p_{r(i)} + p_{r(i)} e^{d_{r(i)}\tau}\big) \leq \eta,$$

or equivalently,

$$\frac{1}{\tau} \sum_{i=1}^{n_r} \ln\big(1 - p_{r(i)} + p_{r(i)} e^{d_{r(i)}\tau}\big) \leq q + \frac{\ln \eta}{\tau}.$$

Note that as $\tau \to \infty$, this constraint recovers the deterministic counterpart. By defining a new "capacity" $\hat{q} := q + (\ln \eta)/\tau$ and new "demands" $\hat{d}_i := (1/\tau) \ln\big(1 - p_{r(i)} + p_{r(i)} e^{d_{r(i)}\tau}\big)$, we can incorporate the chance constraint into the pricing recursion (3).

As an example, suppose a planned route has ten customers, each customer i with $d_i = 2$ and $p_i = 0.5$. Choosing $\tau = 1$, this gives a probabilistic "demand" parameter of $\hat{d}_i \approx 1.43$. To be feasible in the deterministic version of the vehicle capacity constraint, we would require a capacity of 20. If we use $\eta = 0.1$, we obtain via the bound that capacity can be as low as $10 \cdot \hat{d}_i - \ln 0.1 \approx 16.64$ and the route would remain feasible for the chance constraint. This is still an approximation; we can verify by direct calculation that even with a capacity of 14 the route would still be feasible for the chance constraint.

3 Column Generation for VRP-PC

We next study how to price columns for the VRP-PC by approximating a route's expected reduced cost. Later, we provide two solution algorithms based on the VRP-PC column generation model introduced in Section 2.2.

The pricing problem for the VRP-PC master problem in (5) with k-cycle elimination considers the expected cost of each relaxed route $r \in R_k$ and is defined by

$$\min_{r \in R_k} \left\{ \mathbb{E}(c_r) - \sum_{i \in C} \alpha_r^i \rho_i \right\}, \tag{6}$$

where the ρ_i are again dual multipliers. Even though we relax the route set to R_k, the expected cost formula (4) depends on the entire customer visit sequence and increases the subproblem's difficulty. To deal with this additional difficulty, we further relax the expected cost expression and only consider the first k terms,

$$E^k(c_r) := \sum_{\ell=1}^{k} H_r^\ell,$$

where we include arcs that skip up to $k-1$ customers in the sequence. $E^k(c_r)$ is a valid lower bound for $\mathbb{E}(c_r)$ for $k \in \{1, \ldots, n_r\}$, as we show here.

Proposition 1. *For any route $r \in R_1$ and $k \in \{1, \ldots, n_r\}$, the k-term approximation of the expected cost $\mathbb{E}(c_r)$ is monotone non-decreasing in k, i.e., $E^k(c_r) \leq E^{k+1}(c_r)$. Moreover, if $\bar{c}_r$ is the deterministic cost of visiting all customers in r, $E^{n_r}(c_r) = \mathbb{E}(c_r) \leq \bar{c}_r$.*

Proof. $E^{k+1}(c_r) - E^k(c_r) = H_r^{k+1} \geq 0$, so this approximation is monotone non-decreasing. The exact expected cost considers all non-negative terms H_r^ℓ and is thus an upper bound to $E^k(c_r)$. Finally, the deterministic cost is no smaller than the expected value because under the triangle inequality, the cost of each possible realization is always less than or equal to the cost of visiting all customers. □

In the following sections, we provide two algorithms that exploit this lower bound $E^k(c_r)$ on the expected cost of a route r.

3.1 Updating Cost Algorithm

In our first algorithm, which we call the Updating Cost Algorithm (UCA), we obtain a lower bound for the optimal reduced cost of the relaxed master problem (5) by approximately solving subproblem (6) using a non-elementary path relaxation with k-cycle elimination (SPPRC-k), and replacing the exact expected cost $\mathbb{E}(c_r)$ with the approximation $E^k(c_r)$. When solving the SPPRC-k, we can adapt the DP recursion (3) to include expected arc costs corresponding to arcs that skip at most $k-1$ customers, thus allowing us to optimize with respect to $E^k(c_r)$.

We first address how well E^k approximates the true expected cost. Let $\hat{p} := \max_{i \in C} p_i$, $\check{p} := \min_{i \in C} p_i$, and let

$$\hat{c} := \max\Big\{\max_{i,j \in C} c_{ij}, \max_{i \in C}\big\{\max\{c_{0i}, c_{i,n+1}\}/p_i\big\}\Big\}.$$

This last quantity represents the most expensive arc cost in the graph, where we weigh arcs incident to the depot more heavily.

Lemma 2. *Let $y \geq 0$ satisfy $\sum_{r \in R_1} \alpha_r^i y_r = 1$ for each $i \in C$, and let $k \in \{1, \ldots, n\}$. Then*

$$\sum_{r \in R_1} y_r E^k(c_r) \leq \sum_{r \in R_1} y_r \mathbb{E}(c_r) \leq \sum_{r \in R_1} y_r E^k(c_r) + \delta_k,$$

where

$$\delta_k := \hat{c}\hat{p}^2 \sum_{\ell=k+1}^{n} (n - \ell + 2)(1 - \check{p})^{\ell-1}. \tag{7}$$

Although we state the result in terms of the largest route set R_1, the same guarantee applies to any smaller set R_k by taking $y_r = 0$ for $r \notin R_k$.

Proof. The first inequality is a consequence of Proposition 1. To prove the second, we first note that for any route $r \in R_1$,

$$\mathbb{E}(c_r) - E^k(c_r) = \sum_{\ell=k+1}^{n_r} H_r^\ell \leq \hat{c}\hat{p}^2 \sum_{\ell=k+1}^{n_r} (n_r - \ell + 2)(1 - \check{p})^{\ell-1}.$$

Summing over the y_r values, we obtain

$$\sum_{r \in R_1} y_r(\mathbb{E}(c_r) - E^k(c_r)) \leq \hat{c}\hat{p}^2 \sum_{r \in R_1} y_r \sum_{\ell=k+1}^{n_r} (n_r - \ell + 2)(1 - \check{p})^{\ell-1}$$
$$\leq \hat{c}\hat{p}^2 \sum_{\ell=k+1}^{n} (1 - \check{p})^{\ell-1} \sum_{r \in R_1} y_r(n_r - \ell + 2) \leq \delta_k,$$

where the last inequality follows from $\sum_r n_r y_r = \sum_r \sum_i \alpha_r^i y_r = n$ and $\sum_r y_r \geq 1$. □

This result allows us to gauge how closely we approximate our problem's true optimal cost if we use the approximate route cost E^k instead.

Theorem 3. *Suppose we replace $\mathbb{E}$ with E^k in (5), optimize with respect to this objective, and obtain solution y^k. Then*

$$\sum_{r \in R_k} y_r^k E^k(c_r) \leq f(R_k) \leq \sum_{r \in R_k} y_r^k \mathbb{E}(c_r) \leq \sum_{r \in R_k} y_r^k E^k(c_r) + \delta_k.$$

The analogous guarantee holds for the integral case: Suppose R^* is an optimal set of routes for the VRP-PC, and suppose $\bar{R}^k$ is an optimal set with respect to the approximate objective E^k. Then

$$\sum_{r \in \bar{R}^k} E^k(c_r) \leq \sum_{r \in R^*} \mathbb{E}(c_r) \leq \sum_{r \in \bar{R}^k} \mathbb{E}(c_r) \leq \sum_{r \in \bar{R}^k} E^k(c_r) + \delta_k.$$

Proof. The first inequality is a consequence of Lemma 2 and y^k's optimality with respect to E^k. The second follows from y^k's feasibility for (5), and the last from Lemma 2. The same argument can be repeated for the second set of inequalities, restricting the analysis to integer y solutions. □

The theorem implies that if we approximately optimize (5) over routes R_k with objective E^k, we have the *a priori* guarantee that the solution we obtain will be within an additive gap of δ_k from $f(R_k)$. Similarly, if we embed this approximate optimization within a B&P algorithm, we are guaranteed to obtain an integer solution within δ_k of the optimal expected cost. In addition, after carrying out the optimization, we can calculate a tighter *a posteriori* gap by taking the difference of the solution's true expected cost with $\mathbb{E}$, minus its approximate expected cost as measured by E^k.

Corollary 4. *To achieve any desired additive optimality gap $\epsilon \geq 0$ in (5) (and in the integral problem via B&P), it suffices to choose $k = O(\log(n/\epsilon))$.*

Proof. By definition of δ_k, we have

$$\delta_k \leq \hat{c}\hat{p}^2 n \sum_{\ell=k+1}^{n} (1-\check{p})^{\ell-1} \leq \frac{\hat{c}\hat{p}^2 n(1-\check{p})^k}{\check{p}}.$$

Therefore, to guarantee $\delta_k \leq \epsilon$, it suffices for k to satisfy

$$(1-\check{p})^{-k} \geq \frac{\hat{c}\hat{p}^2 n}{\epsilon\check{p}} \quad \Longleftrightarrow \quad k \ln\left(\frac{1}{1-\check{p}}\right) \geq \ln\left(\frac{\hat{c}\hat{p}^2 n}{\epsilon\check{p}}\right). \qquad \square$$

Although this last result shows a logarithmic dependence on n, in practice we find that the δ_k values decrease quite rapidly, so that a small k suffices to provide a very tight gap. Table 1 provides an example of $\delta_k/\hat{c}$ values for $n = 50$ as a function of k and a uniform customer probability. Our

$p \backslash k$	1	2	3	4	5
0.5	12.25	6.00	2.9375	1.4375	0.703125
0.7	10.41	3.06	0.8991	0.26406	0.077517
0.9	4.49	0.44	0.0431	0.00422	0.000413

Table 1: Sample $\delta_k/\hat{c}$ values as a function of k and $p = \check{p} = \hat{p}$, for $n = 50$.

computational results detailed in the next section verify this convergence and also explore the *a posteriori* gap in empirical terms.

```
input: integers K_0 ≤ K, initial set of routes R̃ and associated
  approximate costs E^{K_0}(c_r) for all r ∈ R̃;
set k ← K_0;
while k ≤ K do
    solve restricted master problem using routes R̃ and costs
      E^k;
    solve pricing subproblem as SPPRC-k, obtain new route
      r;
    if E^k(c_r) − Σ_{i∈C} α_r^i ρ_i < 0 then
        include new column in master problem, R̃ ← R̃ ∪ {r};
    else
        k ← k + 1;
        recompute approximate expected costs as E^k(c_r) for
          r ∈ R̃;
    end
end
```

Algorithm 1: Column generation algorithm UCA.

Algorithm (1) details our implementation of this approximate optimization. Instead of starting from the desired approximation precision, the algorithm solves the column generation algorithm sequentially, with increasing precision at every step of the outer loop. Intuitively, we expect pricing subproblems with small k to be easy, and thus to quickly find useful columns in the first steps; for larger values of k, we then obtain a few remaining columns that marginally improve the expected cost. Because the algorithm updates the cost approximation as it progresses, we name it the Updating Cost Algorithm (UCA).

UCA takes as argument an initial set of feasible routes $\tilde{R}$ and two positive integers $K_0 \leq K$; K_0 is the initial value for each route's expected cost approximation and K is the final value used in the approximation for column generation. In each step $k \in \{K_0, \ldots, K\}$, the algorithm searches for routes $r \in R_k$, approximating r's expected cost using $E^k(c_r)$. If the resulting route r has negative reduced cost, we include it in the set $\tilde{R}$ for the master problem; otherwise, we increase k and

recompute the cost of the routes considered in the master using E^k.

UCA begins by generating routes in R_{K_0}, meaning some of these routes may in fact have cycles shorter than or equal to K. Therefore, we cannot claim that the algorithm optimizes the LP relaxation (5) over R_K. We can, however, make a weaker assertion.

Proposition 5. *The value returned by UCA with parameter K is a lower bound for $f(R_K)$.*

Proof. In its final iteration ($k = K$), the algorithm ensures that every route in R_K has non-negative reduced cost with respect to E^K. However, the algorithm can generate routes in its previous iterations, some of which could have cycles of length K or shorter. So UCA produces a solution that is optimal for a superset of R_K, where the inclusion may be strict. □

Finally, we verify that employing UCA within a B&P framework preserves the gap guarantee.

Corollary 6. *Using UCA within a B&P algorithm yields an integer solution with expected cost within an additive gap δ_K of optimal.*

Proof. The proof follows from Theorem 3 and Proposition 5 by noting that if UCA returns an integer solution, this solution must be optimal with respect to E^K. □

We finish this subsection by noting that although UCA has both an *a priori* and an *a posteriori* guarantee, it cannot guarantee a solution with lower expected cost than the solution implied by the deterministic VRP on the same instance. Therefore, in practice we warm start the algorithm with the deterministic solution, which also helps us fathom nodes in the search tree.

3.2 Fixed Cost Algorithm

The motivation for UCA and the use of the cost approximation E^k is the difficulty of the exact pricing problem (6). However, once we generate a particular route r, we can efficiently check its exact expected cost and thus its exact reduced cost. This motivates a second algorithm to approximately solve (5) and the VRP-PC, in which we include routes in the restricted master problem with their exact expected costs; because this second algorithm always keeps routes' true expected cost (instead of updating an approximation), we call it the Fixed Cost Algorithm (FCA).

Algorithm 2 details FCA. We again use two parameters $K_0 \leq K$ and increase the precision of the pricing problem and the expected cost approximation E^k for $k \in \{K_0, \ldots, K\}$; however, in

this case we only add routes if their exact reduced cost (measured with the exact expected cost) is negative. This has the benefit of including columns in the master with their exact objective value, but the disadvantage that we may have an inconclusive pricing outcome, where a route with negative approximate reduced cost in fact has non-negative reduced cost. Such an inconclusive outcome triggers an increase in the pricing precision until we reach K, at which point the algorithm terminates.

```
input: integers K_0 ≤ K, initial set of routes R̃ with expected
  costs 𝔼;
set k ← K_0;
while k ≤ K do
    solve restricted master problem with routes R̃ and costs
      𝔼;
    solve pricing subproblem as SPPRC-k with approximate
      cost E^k, obtain new route r;
    if E^k(c_r) − Σ_{i∈C} α_r^i ρ_i ≥ 0 then
        terminate;
    else if 𝔼(c_r) − Σ_{i∈C} α_r^i ρ_i < 0 then
        include new column in master problem, R̃ ← R̃ ∪ {r};
    else
        k ← k + 1;
    end
end
```

Algorithm 2: Column generation algorithm FCA.

Theorem 7. *The non-basic routes in the solution produced by algorithm FCA have reduced cost bounded below by* $-\delta_K$.

Proof. If the algorithm terminates because the minimum approximate reduced cost is non-negative, the current solution is optimal. Therefore, assume the algorithm terminates because the final route

$\tilde{r}$ priced by the algorithm has an inconclusive reduced cost. That is,

$$E^K(c_{\tilde{r}}) - \sum_{i \in C} \alpha^i_{\tilde{r}} \rho_i < 0 \leq \mathbb{E}(c_{\tilde{r}}) - \sum_{i \in C} \alpha^i_{\tilde{r}} \rho_i,$$

where ρ is an optimal dual solution of the restricted master solved with the current route set $\tilde{R}$. This implies for any route $r \in R_K$ that

$$\sum_{i \in C} \alpha^i_r \rho_i - \mathbb{E}(c_r) \leq \sum_{i \in C} \alpha^i_r \rho_i - E^K(c_r) \leq \sum_{i \in C} \alpha^i_{\tilde{r}} \rho_i - E^K(c_{\tilde{r}}) \leq \mathbb{E}(c_{\tilde{r}}) - E^K(c_{\tilde{r}}) = \sum_{\ell=K+1}^{n_{\tilde{r}}-1} H^\ell_{\tilde{r}} \leq \delta_K,$$

where the second inequality follows because $\tilde{r}$ is the route produced by the approximate pricing problem. □

Like UCA, this algorithm has an *a priori* gap guarantee.

Corollary 8. *Suppose $K_0 = K$ and let y^* be optimal for* (5) *with respect to R_K. The solution produced by FCA has objective value no greater than*

$$\sum_{r \in R_K} y^*_r (\mathbb{E}(c_r) + \delta_K) = f(R_K) + \delta_K \sum_{r \in R_K} y^*_r.$$

As with UCA, a similar guarantee applies when $K_0 < K$, except the set of routes we optimize over may be larger than R_K, as it contains any routes with smaller cycles that the algorithm included in earlier iterations.

Corollary 9. *Using FCA within a B&P algorithm yields an integer solution with expected cost within an additive gap $\delta_K n$ of optimal. The factor of n can be substituted by any known tighter bound on the number of routes used by an optimal solution.*

Proof. The gap given in Corollary 8 depends on an optimal solution of the LP; in B&P, this solution would vary by node, so we can only claim an overall gap that is guaranteed no smaller than the gap at any node. Since we may assume $\sum y_r \leq n$ without loss of optimality, the result follows. If we have an upper bound on $\sum y_r$ that is tighter than n, the same argument applies with this bound. □

Corollary 10. *To achieve any desired additive optimality gap $\epsilon \geq 0$ for the VRP-PC via FCA within B&P, it suffices to take $K = O(\log(n/\epsilon))$.*

Proof. The proof is identical to Corollary 4, except we start with $\delta_K n \leq \epsilon$. □

4 Computational Study

In this section, we test both of our algorithms implementing a proof of concept for the VRP-PC, including a particular hard sequence-dependent cosntraint. We also study the empirical convergence of the expected cost approximation for UCA and FCA, the algorithms' running times, and how their performance is affected by model parameters.

4.1 Instances

We test our algorithms on VRP-PC instances based on the Solomon instances of the VRP with time windows (VRPTW) [44]. When customers may or may not be present, it might not be practical to establish hard time window constraints, as in the VRPTW. Nonetheless, we carry out our experiments with these instances for two reasons: First, time windows are a type of route- or sequence-dependent constraint that is difficult to capture with arc-based formulations, where the corresponding relaxations are known to be weak. Second, the Solomon instance family is a generally accepted benchmark widely used by researchers to test VRP algorithms.

The Solomon instances have 100 customers each and are divided into three categories: C (clustered), R (uniformly distributed) and RC (mix of R and C). Using 5 instances each from type C (C101 to C105) and from R (R101 to R105), we create VRP-PC instances with 15, 25 and 40 customers. Instances with a given number of customers have a different sequence of customers with respect to the original Solomon instance. For example, we create two 40-customer instances from each original Solomon instance, one considering customers 1 to 40 and another with customers 41 to 80. Because our largest instances have 40 customers, we reduce the vehicle capacity from 200 to 80. All instances have the original depot location.

With this procedure, we respectively obtain 60, 40 and 20 "base" deterministic VRPTW instances with 15, 25 and 40 customers. We then tested our B&P implementation on each of these deterministic instances, eliminating two 25-customer and four 40-customer instances we could not

solve to optimality within a 6-hour time limit; this reduction allows us to compare our VRP-PC results against the corresponding deterministic solution, and also lets us focus more on the difficulty increase brought on by the problem's probabilistic nature, rather than on the challenges it inherits from the VRPTW. After this elimination, each remaining base deterministic instance generates three VRP-PC instances, each with a different uniform customer probability $p \in \{0.5, 0.7, 0.9\}$, yielding a total of 342 instances.

4.2 Experiments

We tested both algorithms, UCA and FCA, on each instance with $K_0 = 1$ and $K \in \{1, \ldots, 5\}$, applying a 6-hour time limit. In total, this involves $1,710$ runs of each algorithm on the different instances. We ran the experiments in the Georgia Tech ISyE computing cluster, which uses HT-Condor to manage its jobs, on an Intel Xeon E5-2603 (1.80GHz) machine with up to 10Gb of RAM, and using CPLEX 12.4 as LP solver. As a reference, when we run our B&P implementation on the original deterministic instances with 100 customers, we can solve several instances to optimality (of both type C and R) in a few minutes. This gives further indication that the main computational challenge we face stems from the optimization of expected route cost, and also suggests, as expected, that optimizing the VRP-PC is significantly more difficult than its deterministic counterpart.

In our first set of results, shown in Table 2, we report the average maximum value of K our B&P algorithms were able to solve to optimality and within a 5% and 10% relative gap. (The gap measured here is between the best integer solution and best bound found by the B&P tree.) For example, for instances with 15 customers and probability 0.5, the average largest K value for which our UCA B&P algorithm can report a 0% relative gap within the time limit is 4.45. The corresponding average for FCA is 4.52.

From the table we see the clear impact the number of customers n has on the parameter K we can use when running the algorithms to optimality. For 15 customers, K can be 4 or 5; for 25 customers, K can be about 3 or 4; and for 40 customers K can be 3 and sometimes 2. Similarly, the customer realization probability p affects this average K; as we might expect, the larger the probability, the larger K can be, though there are some exceptions. The choice of K has two separate consequences. First, it helps determine the *a priori* and *a posteriori* additive gap guarantees of solution quality we get from UCA and FCA, since they depend on the number

n	p	UCA			FCA			Total Runs
		0%	5%	10%	0%	5%	10%	
15	0.5	4.45	4.83	4.95	4.52	4.88	4.95	900
	0.7	4.67	4.9	4.97	4.78	4.83	4.98	
	0.9	4.87	4.9	5	4.87	4.93	5	
25	0.5	3.37	4.08	4.68	3.55	4.29	4.71	570
	0.7	3.45	4.11	4.74	3.58	4.16	4.76	
	0.9	3.71	4.11	4.58	3.68	4.37	4.71	
40	0.5	2.5	3.19	4.38	3	3.94	4.19	240
	0.7	2.38	3.69	4.25	2.62	3.88	4.25	
	0.9	2.62	4.38	4.5	2.56	4.19	4.44	

Table 2: Average largest K solved by each algorithm within a certain relative gap.

of terms we consider in the expected value approximation E^K. Second, it impacts the problem's difficulty through the pricing problem, which is solved as an SPPRC-K.

In Figure 1 we depict the average UCA *a priori* and *a posteriori* gap guarantees as a function of K for the three different customer probabilities $p \in \{0.5, 0.7, 0.9\}$. For each value of K and p, we include in the average only those instances we were able to solve to optimality in the B&P algorithm. We report these gaps as relative distances to the optimal solution; the gaps decay quite significantly for higher K values, so we present them in natural logarithmic scale (*i.e.*, as powers of e). For example, for $p = 0.5$ and $K = 3$ the average *a priori* gap is roughly $1/e \approx 37\%$, meaning we can guarantee before running the algorithm that the expected cost of the solution returned by UCA is at worst roughly a third costlier than the optimum, on average.

These results indicate how quickly both guarantees converge to zero as we increase K. For any probability, $K = 4$ or $K = 5$ more than suffice for either algorithm to return a solution with expected cost very close to optimal. Furthermore, for $K \geq 3$ the difference between *a priori* and *a posteriori* gaps is already within 10%-20% or less, with the *a priori* guarantee at worst being about 30% from optimal.

Figure 2 shows the convergence between the UCA and FCA solutions' expected cost and UCA's bound in an absolute scale, as a function of the algorithm parameter K, plotted by customer realization probability. The averages here include only those instances for which we were able to run the B&P algorithm to optimality for all values of K, in order to make the comparison using a fixed set of instances. The figure plots the average of the the UCA optimal value measured with approximation E^K, which is a lower bound on the optimal expected cost; it also plots the UCA and

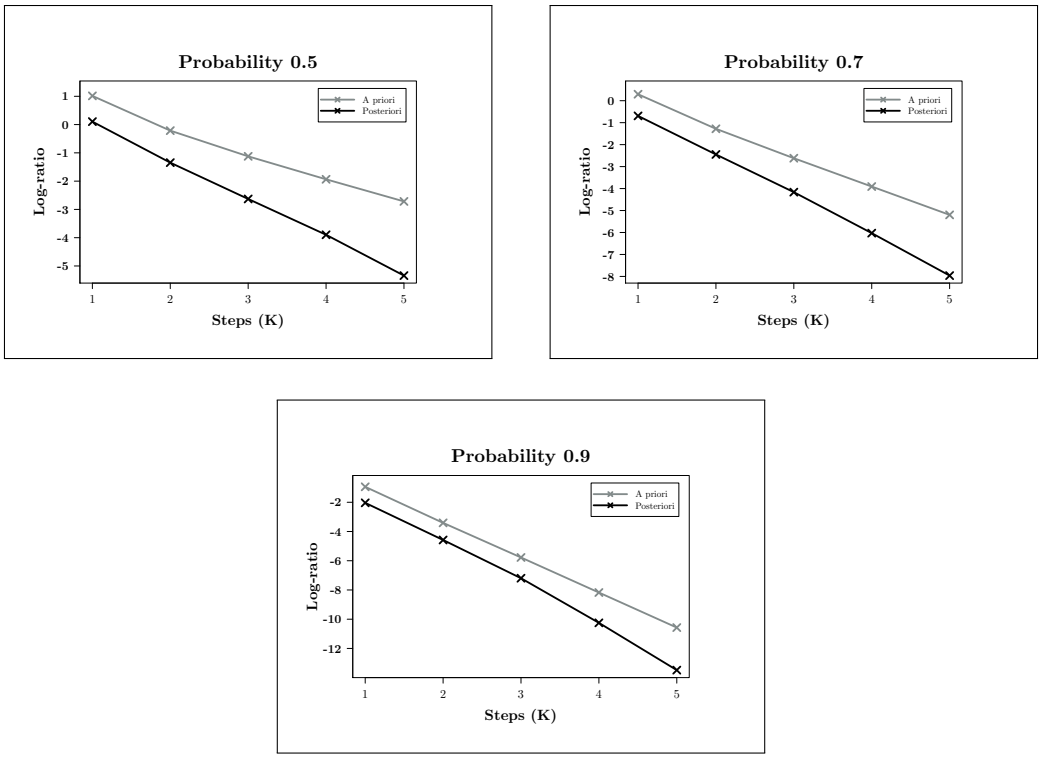

Figure 1: Average *a priori* and *a posteriori* guarantees by probability.

FCA solutions' exact expected cost, UCA + Post and FCA. We observe that the bound provided by UCA converges very fast to the optimum, especially when the customer realization probability is high. Moreover, the expected cost of either algorithm's solution is quite close to the optimum, even for $K = 1$.

Table 3 presents the geometric mean of relative gaps across instances of three solutions: the UCA solution (with exact expected cost), the FCA solution, and the expected cost of the deterministic VRPTW solution that ignores probabilities and minimizes the cost as if all customers will require visits. The gaps are calculated with respect to the best possible lower bound computed by UCA for any value of K, where we include all instances we could solve for that K; this means the number of instances included in an average may vary by value of K. We show results for $p = 0.7$, with similar tables for $p \in \{0.5, 0.9\}$ in the Appendix, and separate results by instance type (C and R); recall that instances of type C have clustered customer locations, while type-R instances have uniformly distributed customer locations.

In all cases, the expected cost of solutions produced by UCA and FCA are within 5% of optimal on average, often much closer, and both are consistently better than the solution given by the

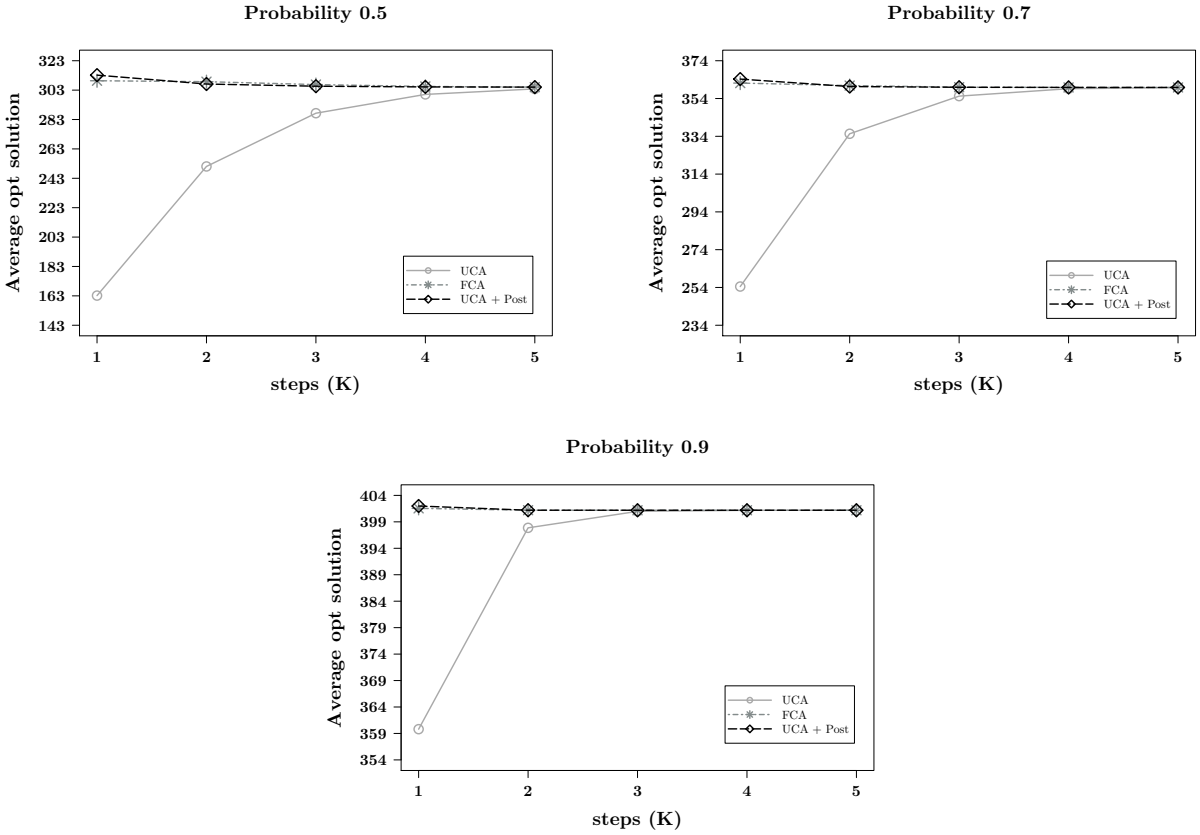

Figure 2: Convergence of UCA lower bound and solutions' expected cost, by probability.

deterministic instance. Either algorithm can produce the better solution on a particular instance; we detect no clear pattern of one producing better solutions than the other, but overall UCA has a slight advantage. Unsurprisingly, the number of customers n impacts the solutions' gap, with bigger instances having larger gaps. More interestingly, the instance type significantly affects the solutions' quality, with gaps for type R on average much tighter than for type C. Unlike the results summarized in Figure 2, here we do not always see a monotonically decreasing gap as K increases, but this is a result of including different sets of instances for different values of K; in general, for a given instance we tend to see better solutions with larger K.

To further explore the benefit of optimizing with respect to expected costs, Figure 3 plots the percentage of instances where UCA and FCA obtain solutions with lower expected cost than the solution of the deterministic VRPTW. For example, when the customer realization probability is 0.5 and $K \geq 4$, both UCA and FCA produce better solutions in about 60% of the tested instances. The plots also emphasize that the realization probability affects this improvement percentage; for larger

Customers	Steps (K)	Type C			Type R		
		$\frac{\text{UCA Exact}}{\text{Best LB}}$	$\frac{\text{FCA}}{\text{Best LB}}$	$\frac{\text{Deter}}{\text{Best LB}}$	$\frac{\text{UCA Exact}}{\text{Best LB}}$	$\frac{\text{FCA}}{\text{Best LB}}$	$\frac{\text{Deter}}{\text{Best LB}}$
15	0	1.55%	1.45%	1.59 %	0.44%	0.53%	0.88 %
	1	0.49%	0.53%		0.39%	0.55%	
	2	0.25%	0.27%		0.41%	0.42%	
	3	0.25%	0.25%		0.41%	0.41%	
	4	0.25%	0.25%		0.41%	0.41%	
25	0	2.32%	2.55%	3.29 %	0.86%	1.09%	1.62 %
	1	2.83%	3.01%		1.15%	1.39%	
	2	2.97%	2.89%		1.09%	1.11%	
	3	2.57%	2.52%		0.97%	0.97%	
	4	2.04%	2.04%		1.01%	1.06%	
40	0	3.93%	3.93%	4.58 %	1.41%	1.41%	1.41 %
	1	4.58%	4.58%		1.11%	1.28%	
	2	4.58%	4.58%		1.06%	1.13%	
	3	4.58%	4.58%		0.07%	0.07%	
	4	3.93%	3.93%		0.07%	0.07%	

Table 3: Average relative gap of solution expected cost for UCA, FCA and deterministic problem, for realization probability 0.7.

probabilities, it is harder for the algorithms to improve on the deterministic solution, presumably because this solution is already close to optimal. In particular, when the probability is 0.9, we only find a better solution roughly one quarter of the time, and this improvement occurs already with $K \geq 2$. In general, we also observe that UCA is slightly better than FCA at improving over the deterministic solution.

Finally, even when the difference in expected cost between the deterministic solution and our algorithms' solutions is not large, the structure of the corresponding solutions changes drastically, and may have operational implications, as discussed next in Section 4.3.

In Table 4 we present the average number of B&B nodes and the average number of routes generated per node for the UCA and FCA algorithms over instances with 15 customers, since those instances are solved closer to optimality. Both algorithms generate a similar magnitude of B&B nodes, but the FCA algorithm shows a more clear increasing trend as p decreases. The average number of routes generated by node is slightly smaller for FCA and for R type instances. Also, for C type instances, this number decreases as the customer realization probability increases.

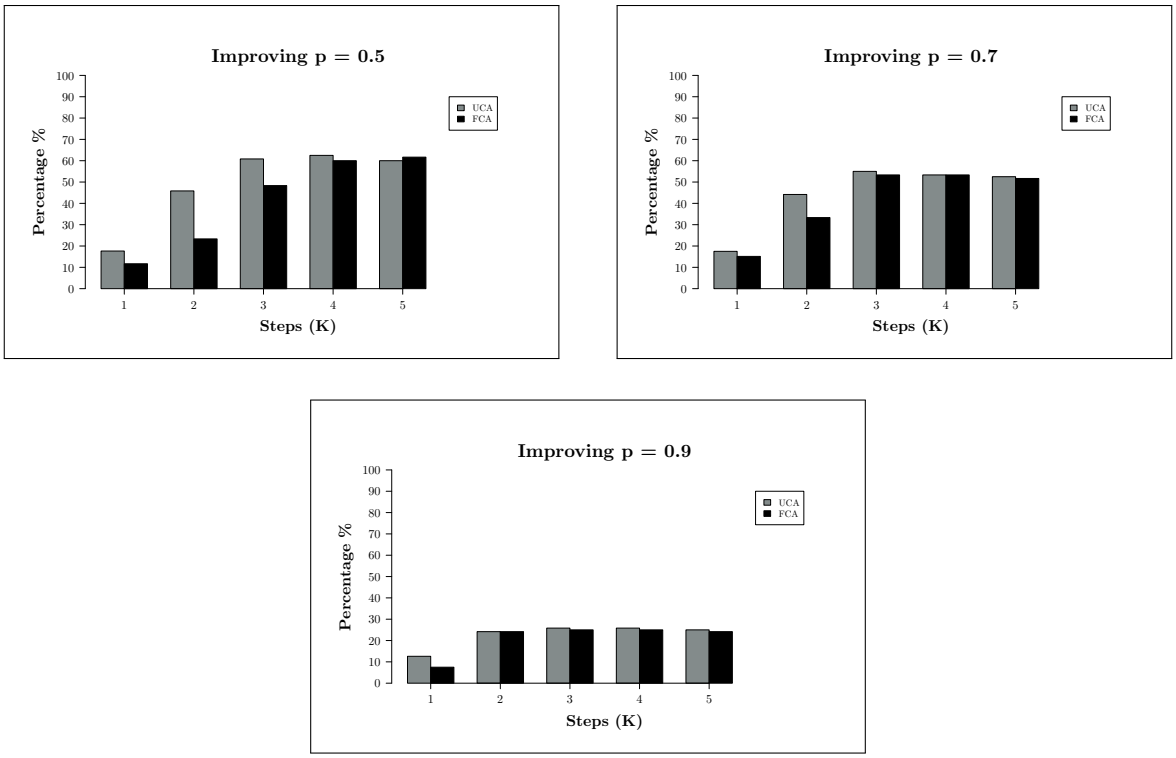

Figure 3: Percentage of instances where UCA and FCA improve on the deterministic solution.

Probability	UCA				FCA			
	Type C		Type R		Type C		Type R	
	Nodes B&P	Avg Routes	Nodes B&P	Avg Routes	Nodes B&P	Avg Routes	Nodes B&P	Avg Routes
0.5	2793.87	71.83	2843.65	42.86	3000.13	53.23	4797.33	38.38
0.7	3580.88	64.19	2356.56	46.79	2434.08	48.74	2678.5	40.09
0.9	2106.52	58.23	2323.48	49.95	1020.77	45.49	1265.13	40.82

Table 4: Average number of B&B nodes and routes generated by UCA and FCA algorithms in instances with 15 customers.

4.3 Empirical Insights

In Figure 4, we plot the UCA solution and a deterministic solution assuming all customers will realize for an instance of type R with $n = 15$ and $p = 0.5$. The expected cost of the solution to the deterministic problem is 314.19, and it is plotted on the left. The expected cost of the UCA solution with $K = 5$ is 304.27, and it is plotted on the right. The difference in expected cost is only about 3%, yet the solutions are very different. Specifically, both solutions use 5 planned routes, but without repeating any, and the UCA solution's planned routes appear inefficient when viewed as deterministic routes; in particular, they include several crossings that the deterministic solution avoids. The results for this instance highlight the potentially counter-intuitive nature of probabilistic routing, where we may plan routes that appear inefficient, but whose expected cost is

in fact lower than routes that appear cheaper.

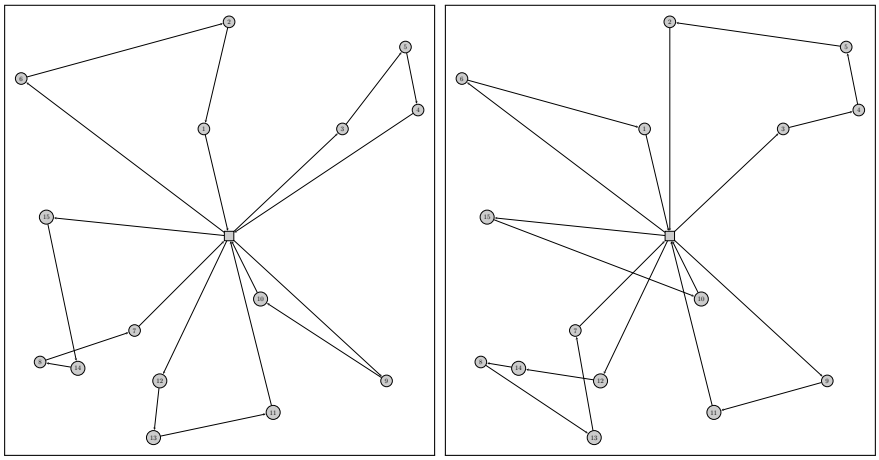

Figure 4: Deterministic (left) and UCA (right, $K = 5$) solutions for sample instance (type R) with $n = 15$ and $p = 0.5$.

We next plot the deterministic and UCA ($K = 3$) solutions for a type-C instance with 40 customers (Figure 5). We observe that some routes are identical in both solutions, while others differ. As in the previous example, routes do not frequently cross for the deterministic solution, while they often do for UCA. Intuitively, the marginal deterministic cost increase generated by routes that cross more frequently is significantly less than the expected cost savings generated by probabilistic customers when they are skipped. Take for example route "c": Its deterministic solution covers two clusters of two and three customers, respectively. The stochastic solution instead covers a cluster of four customers and an isolated customer; if this customer does not show up, the cost of this route decreases significantly, and it is much more likely that one customer is not present, as opposed to the likelihood of two not being present simultaneously. A similar argument can be made for route "f".

5 Conclusions

We have studied the VRP-PC, a broad class of routing problems with probabilistic customers, and proposed a new column generation and B&P framework, including two different algorithms, UCA and FCA. Both circumvent the difficulty of exactly pricing routes by using an approximate expected cost that under-estimates the true expected cost. UCA optimizes with respect to the approximate expected cost, and thus provides an *a posteriori* lower bound to the optimal expected cost, in

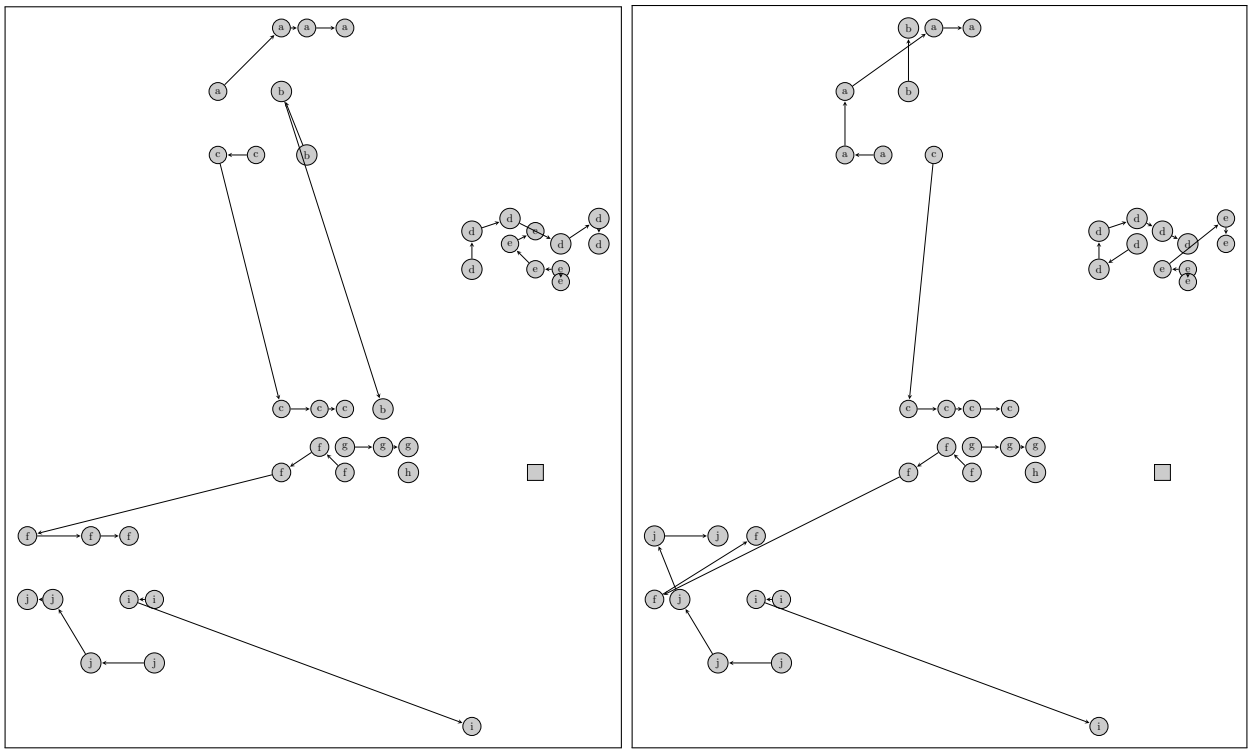

Figure 5: Deterministic (left) and UCA (right, $K = 3$) solutions for sample instance (type C) with $n = 40$ and $p = 0.5$.

addition to giving a solution. FCA uses exact expected costs but may halt when some routes still have small negative reduced cost. Both algorithms have approximate optimality guarantees in the form of *a priori* additive gaps that depend on the precision of the expected cost approximation.

Our computational results suggest the *a posteriori* gap provided by UCA is much tighter than the theoretical *a priori* gap indicates. Furthermore, both gaps decrease quite rapidly as we increase the precision of the expected cost approximation in UCA or FCA; in our instances, an approximation with five steps ($K = 5$) or fewer suffices to get a negligible gap. However, the problem's difficulty is determined also by the number of customers and their realization probability; in particular, when the probabilities are large we can close the gap quickly. More generally, our results also suggest that UCA and FCA produce very good solutions, no more than 5% from optimal and usually much better. Both algorithms improve upon the solution of the deterministic instance in many cases, although the improvement is not always large in terms of relative gap. However, even in these cases the structure of the resulting solution may change significantly.

Our results motivate several questions for future research. One option is to incorporate cutting

planes into our B&P framework with UCA or FCA, which may allow us to increase the size and/or difficulty of instances we can optimize. Another possibility in this vein would be to use different relaxations of the ESPPRC, such as the ng-path relaxations introduced in recent years [1]; however, it is not immediately clear how our analysis or gap guarantees would extend here. Another improvement to our B&P method would be to include a dual stabilization technique; approaches like the *stabilization primal-dual* and *trust region* methods [42] could speed up computation times. An interesting question relates to combining our approach with approaches that optimize other related stochastic routing models. In particular, the use of chance constraints for resource consumption [21] is an interesting area with much potential for future work. Another idea would be to include a recourse action when a key resource is depleted, as in [25]; in this article each customer demand quantity is random and the vehicle is forced to pay a replenishment trip to the depot when it runs out of capacity. The challenge of such an approach would be to include such a recourse action cost within a pricing subproblem. More generally, the broad topic of column generation in probabilistic and *a priori* optimization offers many challenging questions for the research community.

6 Appendix

We present three tables with additional results. Tables 5 and 6 are analogues of Table 3 for $p = 0.5$ and $p = 0.9$, respectively. Table 7 displays average running times for B&P algorithms with UCA and FCA.

Customers	Steps (K)	Type C			Type R		
		$\frac{\text{UCA Exact}}{\text{Best LB}}$	$\frac{\text{FCA}}{\text{Best LB}}$	$\frac{\text{Deter}}{\text{Best LB}}$	$\frac{\text{UCA Exact}}{\text{Best LB}}$	$\frac{\text{FCA}}{\text{Best LB}}$	$\frac{\text{Deter}}{\text{Best LB}}$
	0	2.75%	2.67%		1.79%	2.04%	
	1	1.43%	2.28%		1.24%	1.95%	
15	2	1.01%	1.15%	2.91 %	1.23%	1.19%	2.21 %
	3	0.92%	0.89%		1.14%	1.14%	
	4	0.78%	0.82%		1.28%	1.12%	
	0	4.15%	4.19%		3.17%	3.51%	
	1	4.44%	4.99%		2.66%	3.34%	
25	2	3.83%	4.47%	5.32 %	2.45%	2.78%	3.54 %
	3	3.1%	3.29%		1.6%	1.7%	
	4	1.35%	1.42%		1.65%	1.57%	
	0	6.42%	6.42%		4.49%	4.52%	
	1	7.43%	7.43%		4.22%	4.52%	
40	2	7.25%	7.43%	7.43 %	3.76%	4.14%	4.66 %
	3	7.25%	7.24%		2.29%	2.4%	
	4	5.29%	5.53%		1.79%	1.43%	

Table 5: Average relative gap of solution expected cost for UCA, FCA and deterministic problem, for realization probability 0.5.

Customers	Steps (K)	Type C			Type R		
		$\frac{\text{UCA Exact}}{\text{Best LB}}$	$\frac{\text{FCA}}{\text{Best LB}}$	$\frac{\text{Deter}}{\text{Best LB}}$	$\frac{\text{UCA Exact}}{\text{Best LB}}$	$\frac{\text{FCA}}{\text{Best LB}}$	$\frac{\text{Deter}}{\text{Best LB}}$
	0	0.52%	0.53%		0.11%	0.11%	
	1	0.31%	0.31%		0.04%	0.04%	
15	2	0.31%	0.31%	0.55 %	0.04%	0.04%	0.11 %
	3	0.31%	0.31%		0.04%	0.04%	
	4	0.31%	0.31%		0.04%	0.04%	
	0	0.97%	0.98%		0.43%	0.56%	
	1	2.1%	2.1%		0.51%	0.51%	
25	2	2.1%	2.1%	2.1 %	0.39%	0.39%	0.58 %
	3	1.67%	1.77%		0.11%	0.11%	
	4	0.56%	0.56%		0.11%	0.11%	
	0	2.46%	2.46%		0.77%	0.77%	
	1	3.13%	3.13%		0.77%	0.87%	
40	2	3.13%	3.13%	3.51 %	0.45%	0.45%	0.89 %
	3	1.76%	1.76%		0.49%	0.49%	
	4	2.29%	3.13%		0%	0%	

Table 6: Average relative gap of solution expected cost for UCA, FCA and deterministic problem, for realization probability 0.9.

Customers	Steps (K)	Type C		Type R	
		UCA	FCA	UCA	FCA
15	0	729	361	99	717
	1	1552	878	838	1273
	2	2842	2018	2021	1636
	3	2803	2992	2488	1850
	4	4545	4054	2968	1922
25	0	6909	4736	1497	2205
	1	10622	8822	2945	2951
	2	12714	11462	3700	3571
	3	13715	13803	4989	5021
	4	13794	13431	6691	5983
40	0	12474	8623	8124	3630
	1	19704	16911	8282	7489
	2	20190	20638	9374	9273
	3	20669	21154	10214	9619
	4	21600	20700	10490	9738

Table 7: Average running time for UCA and FCA in seconds.

References

[1] R. Baldacci, A. Mingozzi, and R. Roberti, *New route relaxation and pricing strategies for the vehicle routing problem*, Operations Research **59** (2011), no. 5, 1269–1283.

[2] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance, *Branch-and-price: Column generation for solving huge integer programs*, Operations Research **46** (1998), 316–329.

[3] J. J. Bartholdi III, L. K. Platzman, R. L. Collins, and W. H. Warden III, *A minimal technology routing system for meals on wheels*, Interfaces **13** (1983), no. 3, 1–8.

[4] R. Bent and P. van Hentenryck, *Scenario-based planning for partially dynamic vehicle routing with stochastic customers*, Operations Research **52** (2004), no. 6, 977–987.

[5] O. Berman and D. Simchi-Levi, *Finding the optimal a priori tour and location of a traveling salesman with nonhomogeneous customers*, Transportation Science **22** (1988), no. 2, 148–154.

[6] D. Bertsekas and D. Castanon, *Rollout algorithms for stochastic scheduling problems*, Journal of Heuristics **5** (1999), no. 1, 89–108.

[7] D. Bertsekas, J. N Tsitsiklis, and C. Wu, *Rollout algorithms for combinatorial optimization*, Journal of Heuristics **3** (1997), no. 3, 245–262.

[8] D. J. Bertsimas, *Probabilistic combinatorial optimization problems*, Ph.D. thesis, Massachusetts Institute of Technology, 1988.

[9] D. J Bertsimas, *A vehicle routing problem with stochastic demand*, Operations Research **40** (1992), no. 3, 574–585.

[10] D. J. Bertsimas and L. H. Howell, *Further results on the probabilistic traveling salesman problem*, European Journal of Operational Research **65** (1993), no. 1, 68–95.

[11] D. J. Bertsimas, P. Jaillet, and A. R. Odoni, *A priori optimization*, Operations Research **38** (1990), no. 6, 1019–1033.

[12] J. Branke and M. Guntsch, *Solving the probabilistic tsp with ant colony optimization*, Journal of Mathematical Modelling and Algorithms **3** (2005), no. 4, 403–425.

[13] A. M. Campbell and B. W. Thomas, *Challenges and advances in a priori routing*, The Vehicle Routing Problem: Latest Advances and New Challenges, Springer, 2008, pp. 123–142.

[14] A. M. Campbell and B. W. Thomas, *Probabilistic traveling salesman problem with deadlines*, Transportation Science **42** (2008), no. 1, 1–21.

[15] C. H. Christiansen and J. Lysgaard, *A branch-and-price algorithm for the capacitated vehicle routing problem with stochastic demands*, Operations Research Letters **35** (2007), no. 6, 773–781.

[16] J. Cordeau, G. Laporte, F. Pasin, and S. Ropke, *Scheduling technicians and tasks in a telecommunications company*, Journal of Scheduling **13** (2010), no. 4, 393–409.

[17] J. F. Cordeau, G. Laporte, M. W. P. Savelsbergh, and D. Vigo, *Vehicle routing*, Transportation, handbooks in operations research and management science **14** (2006), 367–428.

[18] G. Desaulniers, J. Desrosiers, and M. M. Solomon, *Column generation*, vol. 5, Springer Science & Business Media, 2006.

[19] M. Desrochers, J. Desrosiers, and M. M. Solomon, *A new optimization algorithm for the vehicle routing problem with time windows*, Operations Research **40** (1992), no. 2, 342–354.

[20] J. Desrosiers and M. E. Lübbecke, *A primer in column generation*, Column Generation, Springer US, 2005, pp. 1–32.

[21] T. Dinh, R. Fukasawa, and J. Luedtke, *Exact algorithms for the chance-constrained vehicle routing problem*, International Conference on Integer Programming and Combinatorial Optimization, Springer, 2016, pp. 89–101.

[22] M. Dror, *Note on the complexity of the shortest path models for column generation in vrptw*, Operations Research **42** (1994), no. 5, 977–978.

[23] F. Errico, G. Desaulniers, M. Gendreau, and L.-M. Rousseau, *A priori optimization with recourse for the vehicle routing problem with hard time windows and stochastic service times*, European Journal of Operational Research **249** (2016), 55–66.

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

[25] M. Gendreau, G. Laporte, and R. Séguin, *An exact algorithm for the vehicle routing problem with stochastic demands and customers*, Transportation Science **29** (1995), no. 2, 143–155.

[26] M. Gendreau, G. Laporte, and R. Seguin, *Stochastic vehicle routing*, European Journal of Operational Research **88** (1996), no. 1, 3–12.

[27] B. L. Golden, S. Raghavan, and E. A. Wasil (eds.), *The Vehicle Routing Problem: Latest Advances and New Challenges*, Springer, 2008.

[28] J. C Goodson, B. W Thomas, and J. W Ohlmann, *A rollout algorithm framework for heuristic solutions to finite-horizon stochastic dynamic programs*, European Journal of Operational Research **258** (2017), no. 1, 216–229.

[29] H. Hashimoto, S. Boussier, M. Vasquez, and C. Wilbaut, *A grasp-based approach for technicians and interventions scheduling for telecommunications*, Annals of Operations Research **183** (2011), no. 1, 143–161.

[30] S. Irnich and D. Villeneuve, *The shortest-path problem with resource constraints and k-cycle elimination for $k \geq 3$*, INFORMS Journal on Computing **18** (2006), 391–406.

[31] P. Jaillet, *Probabilistic traveling salesman problems*, Ph.D. thesis, Massachusetts Institute of Technology, 1985.

[32] P.. Jaillet, *A priori solution of a traveling salesman problem in which a random subset of the customers are visited*, Operations Research **36** (1988), no. 6, 929–936.

[33] P. Jaillet, *Analysis of probabilistic combinatorial optimization problems in euclidean spaces*, Mathematics of Operations Research **18** (1993), no. 1, 51–70.

[34] M. Klapp, *Dynamic Optimization for Same-Day Delivery Operations*, Ph.D. thesis, Georgia Institute of Technology, 2016.

[35] M. Klapp, A. Erera, and A . Toriello, *The dynamic dispatch waves problem for same-day delivery*, European Journal of Operational Research **271** (2018), 519–534.

[36] M. Klapp, A. Erera, and A. Toriello, *The one-dimensional dynamic dispatch waves problem*, Transportation Science **52** (2016), no. 2, 402–415.

[37] G. Laporte and F. Louveaux, *The integer l-shaped method for stochastic integer programs with complete recourse*, Operations Research Letters **13** (1993), no. 3, 133–142.

[38] G. Laporte, F. Louveaux, and H. Mercure, *The vehicle routing problem with stochastic travel times*, Transportation Science **26** (1992), no. 3, 161–170.

[39] G. Laporte, F. V. Louveaux, and H. Mercure, *A priori optimization of the probabilistic traveling salesman problem*, Operations Research **42** (1994), no. 3, 543–549.

[40] A. Larsen, O. B. G. Madsen, and M. M. Solomon, *The a priori dynamic traveling salesman problem with time windows*, Transportation Science **38** (2004), no. 4, 459–472.

[41] T. Leipälä, *On the solutions of stochastic traveling salesman problems*, European Journal of Operational Research **2** (1978), no. 4, 291–297.

[42] M. E Lubbecke and J. Desrosiers, *Selected topics in column generation*, Operations Research **53** (2005), no. 6, 1007–1023.

[43] V. Pillac, C. Gueret, and A. Medaglia, *A parallel matheuristic for the technician routing and scheduling problem*, Optimization Letters **7** (2013), no. 7, 1525–1535.

[44] M. M. Solomon, *Algorithms for the vehicle routing and scheduling problems with time window constraints*, Operations Research **35** (1987), no. 2, 254–265.

[45] L. Talarico, K. Sörensen, and J. Springael, *Metaheuristics for the risk-constrained cash-in-transit vehicle routing problem*, European Journal of Operational Research **244** (2015), no. 2, 457–470.

[46] H. Tang and E. Miller-Hooks, *Approximate procedures for probabilistic traveling salesperson problem*, Transportation Research Record: Journal of the Transportation Research Board (2004), no. 1882, 27–36.

[47] D. Taş, M. Gendreau, N. Dellaert, T. Van Woensel, and A. De Kok, *Vehicle routing with soft time windows and stochastic travel times: A column generation and branch-and-price solution approach*, European Journal of Operational Research **236** (2014), no. 3, 789–799.

[48] F. A. Tillman, *The multiple terminal delivery problem with probabilistic demands*, Transportation Science **3** (1969), no. 3, 192–204.

[49] P. Toth and D. Vigo, *Vehicle routing: Problems, methods, and applications*, vol. 18, SIAM, 2014.

[50] M. W. Ulmer, D. C. Mattfeld, M. Hennig, and J. C. Goodson, *A rollout algorithm for vehicle routing with stochastic customer requests*, Logistics Management, Springer, 2016, pp. 217–227.

[51] S. Voccia, A. M. Campbell, and B. W. Thomas, *The probabilistic traveling salesman problem with time windows*, EURO Journal on Transportation and Logistics (2012), 1–19.

[52] S. Voccia, A.M. Campbell, and B.W. Thomas, *The same-day delivery problem for online purchases*, published online on Transportation Science (2015), 1–18.