Skip or Insert? A Priori Optimization for the Vehicle Routing Problem with Time Windows and Stochastic Customers

Yulin Han[a,*], Hande Yaman[a]

[a] ORSTAT, Faculty of Economics and Business, KU Leuven, Leuven, 3000, Belgium

Abstract

We study an extension of the vehicle routing problem with time windows by incorporating stochastic customers, i.e., ad-hoc service requests. The uncertainty in stochastic customers is captured through scenarios. Two a priori optimization approaches, a classical and a new one lead to two different problems, both of which are modeled as scenario-based two-stage stochastic programs. While the classical approach plans routes considering both regular and stochastic customers in the first stage and skips the stochastic customers who do not request a visit in the second stage, the new approach plans routes for regular customers in the first stage and inserts realized stochastic customers into the planned routes in the second stage. We prove that the new approach is not more costly than the classical approach. Then to solve the resulting problem, we propose deterministic equivalent formulations, strengthen them with valid inequalities and develop a Cut-and-Branch algorithm and an Integer L-shaped algorithm. The latter incorporates tailored optimality cuts based on the problem structure and several acceleration techniques. We also use a similar Cut-and-Branch algorithm to the solve the problem for the classical approach. Computational experiments are conducted to evaluate the effectiveness of the valid inequalities and the exact methods as well as the performance of the two a priori approaches, and to analyze the influence of insertion path flexibility on solution quality.

Keywords: vehicle routing; time windows; stochastic customers; two-stage stochastic programming; valid inequalities; cut-and-branch; Benders decomposition; integer L-shaped method

*Corresponding author.

Email addresses: yulin.han@kuleuven.be (Yulin Han), hande.yaman@kuleuven.be (Hande Yaman)

1. Introduction

The Field Service Management (FSM) market is projected to grow at a compound annual growth rate of 13.7% between 2024 and 2032 (SNS Insider, 2023), reflecting its increasing importance across service-driven industries. Field service involves dispatching technicians or service personnel to customer locations for the fulfillment of service tasks. It is a critical operational component in a variety of sectors, including manufacturing, information technology (IT), healthcare, and more (Oracle, 2025). In the manufacturing sector, for example, technicians are routinely dispatched to perform on-site maintenance and repairs on complex machinery, helping to ensure operational continuity and to minimize production downtime (ISG, 2025). In the IT domain, field service plays a pivotal role in the deployment, configuration, and troubleshooting of enterprise hardware and networking infrastructure at customer locations (All I.T., 2025). In the healthcare sector, rising demand for home-based care, driven by an aging population and a shift towards patient-centered care, has increased the reliance on timely and reliable field service delivery (Gotadki, 2021).

Despite its importance, field service execution is fraught with operational uncertainties, which complicate decision making and hinder service efficiency. Key sources of uncertainty include ad-hoc service requests, unpredictable service durations, fluctuating travel times, and so on. Among these, the arrival of unexpected service requests is particularly disruptive, as it can invalidate preplanned service routes and strain available resources. Effectively managing such uncertainty is thus essential to improving the responsiveness, reliability, and cost-efficiency of field service operations.

Existing research on the Vehicle Routing Problem (VRP) with ad-hoc service requests has mainly evolved along two methodological paradigms: stochastic (Gendreau et al., 2014) and dynamic approaches (Bektaş et al., 2014). These paradigms differ primarily in when and how information about ad-hoc requests is incorporated. The stochastic approach exploits probabilistic information to preplan routes before operations begin. In contrast, the dynamic approach updates routing decisions in real time, incorporating ad-hoc requests as they occur during execution.

Aligned with the stochastic modeling paradigm yet distinct from existing studies, this work investigates a VRP considering two types of customers. The first type, referred to as *regular customers*, represents deterministic service requests with specified visiting time windows. The second type consists of ad-hoc service requests whose actual need for service is uncertain at the route-planning stage. We refer to these ad-hoc service requests as *stochastic customers*. Their availability is modeled through a finite set of scenarios, each representing a possible realization of customer presence. This work aims to design vehicle routes that serve all regular customers while explicitly accounting for stochastic customers. The resulting problem is termed the *Vehicle Routing Problem with Time Windows and Stochastic Customers* (VRPTW-SC).

For the stochastic VRP, three primary modeling paradigms are commonly employed (Gendreau et al., 2014): (i) a priori optimization, where routes are designed in advance and recourse actions are executed according to a predefined policy once the uncertainty is realized; (ii) re-optimization, where routes are redesigned after the realization of uncertain information; and (iii) Chance-Constrained Programming (CCP), which incorporates probabilistic constraints to ensure solution feasibility with a specified confidence level. To solve the VRPTW-SC, we adopt an a priori optimization paradigm, motivated by the following considerations: the maximum route duration in this work is modeled as a hard constraint that cannot be violated, and the re-optimization often leads to increased computational and operational complexity, potentially undermining schedule reliability and causing driver confusion due to frequent route changes.

Classical a priori optimization approaches for routing problems with stochastic customers (Laporte et al., 1994; Wissink, 2023; Lagos et al., 2023) design the a priori route(s) covering all potential customers and subsequently adjust these route(s) by skipping absent customers once uncertainty is realized. In addition to this approach, we introduce a novel a priori optimization framework that plans routes exclusively for regular customers and inserts or outsources the stochastic customers after the resolution of uncertainty. In this study, we develop models and methods to solve the two resulting scenario-based two-stage stochastic VRPTWs. The first problem, called VRPTW-SC-Insert, constructs vehicle routes in the first stage to serve all regular customers exactly once within their time windows, while ensuring that each vehicle returns to the depot within the maximum allowable route duration. In the second stage, stochastic customers that are realized in a given scenario are either inserted into preplanned routes or outsourced at a predefined cost through recourse actions. The objective is to minimize the expected routing and outsourcing costs. The second problem, called VRPTW-SC-Skip, pursues the same objective by planning first-stage routes for both regular and stochastic customers, deciding for each stochastic customer whether to provide service directly or outsource them at a cost. In the second stage, routes are adapted to each scenario by skipping the customers that do not require a visit. In both problems, we assume that routes that visit only stochastic customers are not allowed.

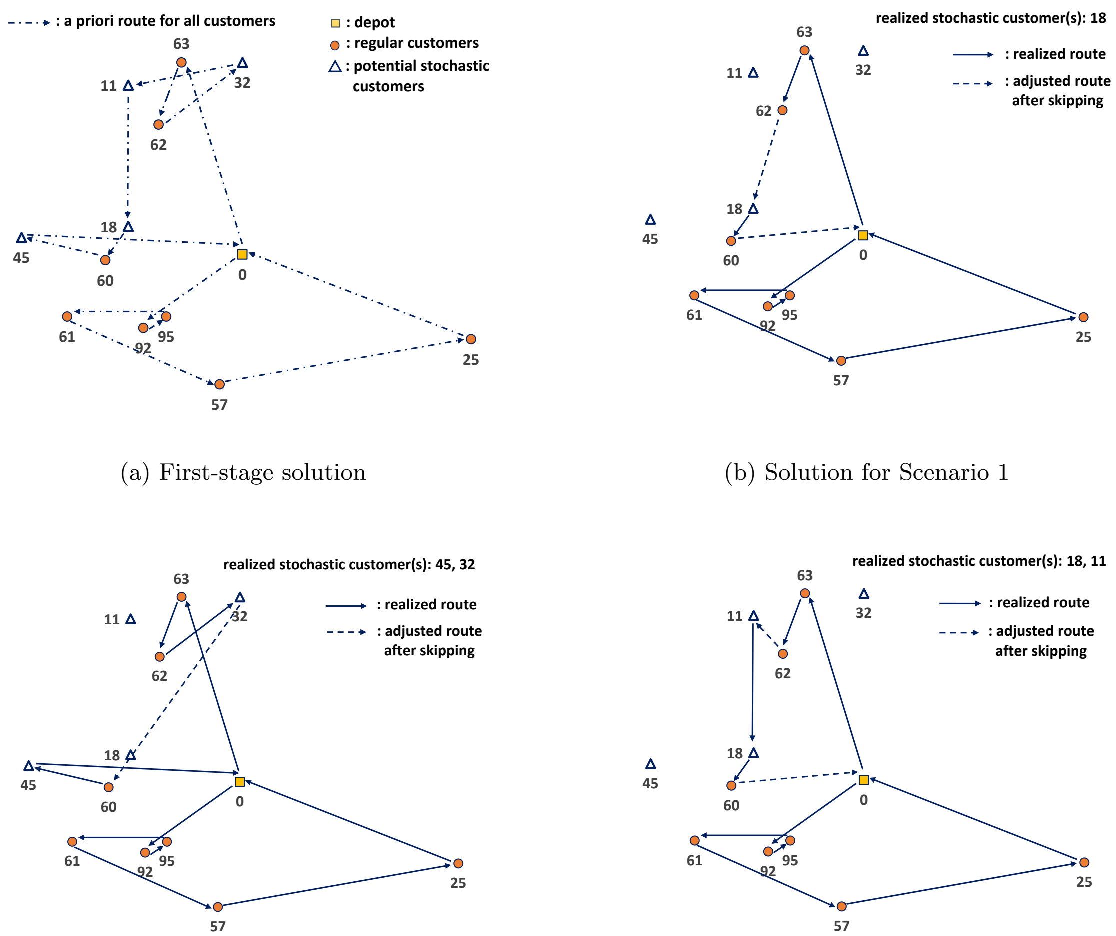
To illustrate the differences between the two a priori optimization frameworks, we employ an example derived from instance R101 (Solomon, 1987), consisting of 8 regular customers, 4 stochastic customers and 3 scenarios. For each framework, the solutions presented in Figures 1 and 2 are obtained from the corresponding two-stage stochastic programming models. The results show that VRPTW-SC-Insert achieves a lower expected total cost of 235, compared with 237.67 for VRPTW-SC-Skip. This improvement stems from the increased flexibility of deferring routing decisions for stochastic customers to the recourse stage and allowing different insertion possibilities for individual scenarios.

The main contributions of this study are threefold:

- We introduce an extension of the VRPTW that accounts for stochastic customers and formulate it as scenario-based two-stage stochastic programs under two different a priori optimization frameworks. We prove that the optimal value of VRPTW-SC-Insert is not larger than the one of VRPTW-SC-Skip. We develop mixed-integer linear programming (MILP) formulations that serve as deterministic equivalents of the stochastic models. To enhance the strength of the linear programming (LP) relaxations, we present lifted versions of the *Short Infeasible Path Constraints* (SIPCs) and use them in conjunction with the *Subtour Elimination Constraints* (SECs).

- To solve the VRPTW-SC-Insert exactly, we propose two exact algorithms: a Cut-and-Branch (C&B) algorithm and an Integer L-shaped (ILS) algorithm. The ILS algorithm incorporates an adapted optimality cut based on the problem structure and is further accelerated through several enhancement strategies to improve computational performance. Similar algorithms can be developed for VRPTW-SC-Skip. We use such a C&B algorithm in our computational experiments.

- We conduct comprehensive computational experiments to evaluate the effectiveness of the introduced valid inequalities (VIs), compare the performance of the proposed exact solution approaches, compare the performance of the optimal solutions of



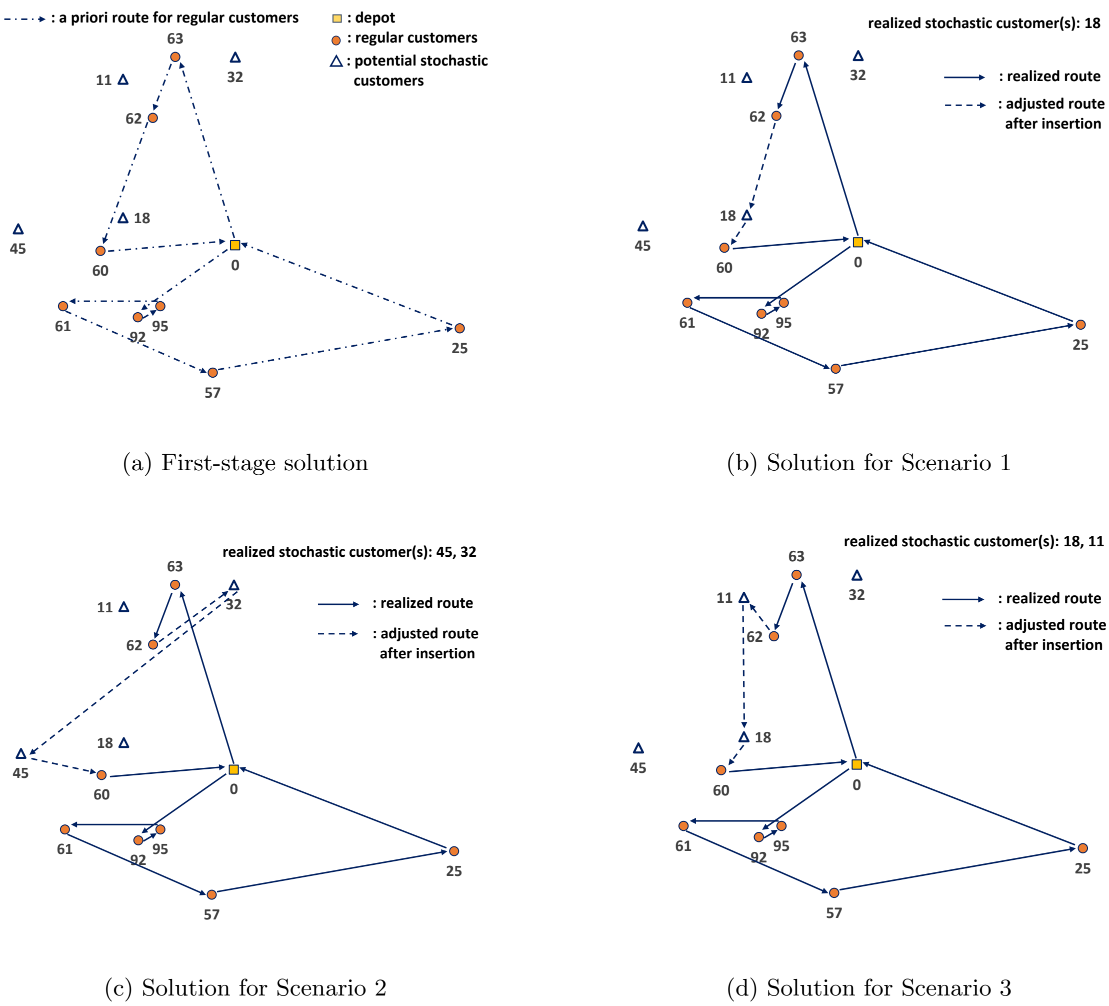
(a) First-stage solution

(b) Solution for Scenario 1

(c) Solution for Scenario 2

(d) Solution for Scenario 3

Figure 1: Optimal solution of VRPTW-SC-Skip (expected total cost: 237.67)



(a) First-stage solution

(b) Solution for Scenario 1

(c) Solution for Scenario 2

(d) Solution for Scenario 3

Figure 2: Optimal solution of VRPTW-SC-Insert (expected total cost: 235)

VRPTW-SC-Insert and VRPTW-SC-Skip, and examine the impact of insertion path flexibility on solution quality for VRPTW-SC-Insert.

The remainder of the paper is organized as follows. Section 2 reviews the relevant literature. Section 3 formally defines the VRPTW-SC and presents deterministic equivalent formulations, along with the VIs designed to strengthen them. Section 4 details the exact solution algorithms. Section 5 reports and analyzes the computational results. Finally, Section 6 concludes the paper and outlines possible directions for future research.

2. Related literature

This section reviews studies on related problems, namely routing problems with probabilistic customers, and the Vehicle Routing Problem with Time Windows (VRPTW), with a focus on exact solution methods.

2.1. Routing problems with probabilistic customers

The Probabilistic Traveling Salesman Problem (PTSP), pioneered by Jaillet (1985, 1988), addresses single-vehicle routing under customer presence uncertainty. Laporte et al. (1994) formulated the problem as a two-stage stochastic program using a classical a priori optimization approach, where each customer has a known presence probability. They proposed the first exact solution algorithm, a Branch-and-Cut (B&C) algorithm, for this problem. More recently, Wissink (2023) extended the PTSP to account for correlations in customer presence, also adopting a classical a priori optimization framework. They developed an exact algorithm based on the ILS method, along with two heuristic methods.

The Vehicle Routing Problem with Probabilistic Customers (VRPPC) generalizes the PTSP to multiple vehicles. Early work by Waters (1989) presented two two-step approaches based on an MILP model: an a priori optimization with two recourse policies, i.e., following the planned routes or skipping absent customers, and a re-optimization, analyzing their performance in terms of traveled distance and fleet size. Lagos et al. (2023) developed Branch-and-Price (B&P) algorithms for the problem using two modeling paradigms: a classical a priori optimization with Bernoulli-distributed customer availability, and a CCP model allowing probabilistic violations of vehicle capacity constraints. Özarik et al. (2023) studied a VRPPC variant, where customer presence is revealed upon vehicle arrival and failed visits trigger same-day revisits. They formulated an MILP model with penalties for failed attempts and recourse-based revisits, benchmarking against heuristic solutions. In a more complex setting, Gendreau et al. (1995) considered both stochastic customers and demands, where each customer has a known probability of being present and a random demand. They modeled the problem as a two-stage stochastic program with recourse actions of skipping absent customers and returning to the depot upon capacity exhaustion, solving it via an ILS method.

2.2. Vehicle routing problem with time windows

The B&P algorithm for the VRPTW, introduced by Desrochers et al. (1992), is a dominant exact solution method. It combines column generation (CG) with Branch-and-Bound (B&B), modeling the problem as a set partitioning formulation (SPF). The pricing subproblem (SP) is typically expressed as an elementary shortest-path problem with resource constraints (ESPPRC). When cutting planes are incorporated, the approach becomes a Branch-and-Cut-and-Price (BCP) algorithm. Kohl et al. (1999) developed the first BCP method for the VRPTW by introducing general k-path VIs (especially 2-path cuts) within a CG scheme. The ESPPRC is the main computational bottleneck of CG. To accelerate its solution, several methods have been proposed, including bounded bi-directional dynamic programming (DP) (Righini and Salani, 2006), state-space augmenting strategies (Boland et al., 2006), and decremental state-space relaxation (Righini and Salani, 2008). To mitigate degeneracy and convergence issues, Rousseau et al. (2007) proposed interior point stabilization methods.

Desaulniers et al. (2008) enhanced the BCP algorithm by incorporating a tabu search heuristic for the ESPPRC, partially relaxing elementary constraints, and generalizing k-path VIs. Jepsen et al. (2008) introduced subset-row VIs and a modified dominance criterion for the label-setting algorithm. Spoorendonk and Desaulniers (2010) applied clique VIs to further strengthen the SPF. Irnich et al. (2010) proposed an arc elimination technique based on path-reduced costs to reduce the network size without compromising optimality. Baldacci et al. (2011) introduced the ng-route relaxation and a column-and-cut generation strategy and Pecin et al. (2017) developed a state-of-the-art BCP framework incorporating limited-memory rank-1 VIs and a new family of strong elementary VIs.

The Branch-and-Cut (B&C) algorithm is another popular exact solution method for the VRPTW, which iteratively adds VIs to strengthen the LP relaxation and uses B&B to enforce integrality. Various VIs have been proposed, including comb inequalities, incompatible pair/path inequalities (Bard et al., 2002), reachability cuts (Lysgaard, 2006), and lifted/local reachability cuts (Avella et al., 2013). Letchford and Salazar-Gonzalez (2006) modeled the VRPTW as a two-commodity flow formulation and derived VIs by projecting the formulation onto its subspace. Kallehauge et al. (2007) proposed binary arc formulations, where time windows are modeled using path inequalities. They also proposed a new class of strengthened path VIs and then developed a B&C algorithm.

Other exact approaches for solving the VRPTW also exist. Kohl and Madsen (1997) applied Lagrangian relaxation on assignment constraints, solving the master problem (MP) with sub-gradient and bundle methods, and SPs via 2-cycle elimination. Kallehauge et al. (2006) extended this approach into an Lagrangian BCP framework, where a stabilized cutting-plane algorithm for the Lagrangian dual problem is embedded in a B&B tree and both SECs and 2-path cuts are introduced to strengthen the MP. Additionally, Rousseau et al. (2004) proposed a constraint programming-based CG solution method for the VRPTW.

3. Problem descriptions, mathematical formulations and valid inequalities

In this section, we first describe the problems and the input parameters. Then we give two deterministic equivalent formulations (DEFs) for VRPTW-SC-Insert. For VRPTW-SC-Skip, we give a single formulation as the other one can be derived in a similar way. Finally, we introduce valid inequalities to strengthen the formulations for both variants.

3.1. Problem descriptions and notations

The VRPTW-SC involves two types of customers: regular customers and stochastic customers. Let N denote the set of regular customers, whose requests are fully known at the time of route planning. Each regular customer $i \in N$ is associated with a time window $[e_i, l_i]$ within which the customer must be visited. In contrast, stochastic customers are uncertain during the planning phase. To capture this uncertainty, we use a finite set of scenarios, denoted by Ω, where each scenario $\omega \in \Omega$ represents a possible realization of stochastic customers and occurs with probability ρ_ω, such that $\rho_\omega > 0$ and $\sum\limits_{\omega\in\Omega} \rho_\omega = 1$. In scenario ω, let N^ω denote the set of stochastic customers requiring service. For each stochastic customer $i \in N^\omega$, a cost β_i is incurred if the customer is not served directly but instead outsourced.

Let node 0 denote the depot, and define the node set as $N_0 = N \cup \{0\}$. For each scenario $\omega \in \Omega$ and $i, j \in N_0 \cup N^\omega$ with $i \neq j$, let t_{ij} and c_{ij} represent the travel time and travel cost, respectively, from node i to node j. We assume that t_{ij} includes service time at i, and the service time at depot is 0. Additionally, both travel times and costs are assumed to satisfy the triangle inequality. A fleet of unlimited homogeneous vehicles, initially located at the depot, is dispatched at time 0 and must return to the depot by a maximum route duration T. We let $e_0 = 0$ and $l_0 = T$. We also define the arc set $A = \{(i,j) : i,j \in N_0 : i \neq j, e_i + t_{ij} \leq l_j\}$ in which we include the arcs (i,j) on which a vehicle can feasibly travel from node i to node j within the time window constraints and we let $G = (N_0, A)$.

3.2. DEFs for VRPTW-SC-Insert

In VRPTW-SC-Insert, vehicle routes are constructed in the first stage to serve all regular customers exactly once within their time windows, while ensuring that all vehicles return to the depot by time T. In the second stage, realized stochastic customers are handled through one of two recourse actions: (i) insertion into the planned routes, or (ii) outsourcing at a predefined cost. The overall objective is to minimize the total routing cost for regular customers plus the expected recourse cost across all scenarios - comprising both the routing cost for served stochastic customers and the outsourcing cost for unserved ones.

We present two formulations for the VRPTW-SC-Insert: a Miller-Tucker-Zemlin formulation (MTZF) inspired by Miller et al. (1960), and a single-commodity flow-based formulation (SCFF) inspired by Gavish and Graves (1979).

Let $p = (k_1, k_2, \ldots, k_m)$ be a path consisting solely of a subset of stochastic customers realized in a given scenario. The total cost and time for traveling from i to j via p are given respectively by $c^p_{ij} = c_{i,k_1} + \sum_{l=1}^{m-1} c_{k_l,k_{l+1}} + c_{k_m,j}$ and $t^p_{ij} = t_{i,k_1} + \sum_{l=1}^{m-1} t_{k_l,k_{l+1}} + t_{k_m,j}$. The path p is considered feasible for insertion into arc $(i,j) \in A$ only if both temporal and cost feasibility conditions are satisfied, that is, $e_i + t^p_{ij} \leq l_j$ and $c^p_{ij} - c_{ij} < \sum\limits_{k \in p} \beta_p$. Let P^{ω}_{ij} denote the set of all feasible paths composed exclusively of stochastic customers realized under scenario ω that can be inserted into arc $(i,j) \in A$.

The decision variables used in both formulations are defined as follows:

$x_{ij} = 1$ if the arc $(i,j) \in A$ is traversed, and 0 otherwise.

$y^{p\omega}_{ij} = 1$ if a path $p \in P^{\omega}_{ij}$ is inserted into $(i,j) \in A$ under scenario ω when $x_{ij} = 1$, and 0 otherwise.

For simplicity, we define the following notations. For $i \in N$, $\delta^+(i)$ $(\delta^-(i))$ denotes the set of arcs in G leaving (entering) i. For any $S \subseteq N$, $A(S)$ denotes the set of arcs with both endpoints in S, whereas $\delta^+(S)$ ($\delta^-(S)$) denotes the set of arcs leaving (entering) S.

3.2.1. Miller-Tucker-Zemlin formulation

To derive the MTZF for the VRPTW-SC, we define the continuous variable a_i^ω denoting the arrival time at $i \in N$ in scenario $\omega \in \Omega$. This formulation is as follows:

$$\min \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{\omega\in\Omega} \rho_\omega \left(\sum_{(i,j)\in A} \sum_{p\in P_{ij}^\omega} (c_{ij}^p - c_{ij}) y_{ij}^{p\omega} + \sum_{k\in N^\omega} \beta_k (1 - \sum_{(i,j)\in A} \sum_{p\in P_{ij}^\omega : k\in p} y_{ij}^{p\omega}) \right) \tag{1a}$$

subject to:

$$\sum_{j:(j,i)\in\delta^-(i)} x_{ji} = 1, \quad i \in N \tag{1b}$$

$$\sum_{j:(i,j)\in\delta^+(i)} x_{ij} = 1, \quad i \in N \tag{1c}$$

$$\sum_{(i,j)\in A} \sum_{p\in P_{ij}^\omega : k\in p} y_{ij}^{p\omega} \leq 1, \quad \omega \in \Omega, k \in N^\omega \tag{1d}$$

$$\sum_{p\in P_{ij}^\omega} y_{ij}^{p\omega} \leq x_{ij}, \quad \omega \in \Omega, (i,j) \in A \tag{1e}$$

$$a_j^\omega \geq a_i^\omega + t_{ij}x_{ij} + \sum_{p\in P_{ij}^\omega} (t_{ij}^p - t_{ij}) y_{ij}^{p\omega} - (l_i - e_j)(1 - x_{ij}), \quad \omega \in \Omega, (i,j) \in A : i,j \neq 0 \tag{1f}$$

$$\begin{aligned} a_j^\omega \geq & \sum_{i:(i,j)\in\delta^-(j)} \max\{e_j, e_i + t_{ij}\} x_{ij} + \sum_{i:(i,j)\in\delta^-(j), e_j \leq e_i + t_{ij}} \sum_{p\in P_{ij}^\omega} (t_{ij}^p - t_{ij}) y_{ij}^{p\omega} \\ & + \sum_{i:(i,j)\in\delta^-(j), e_j > e_i + t_{ij}} \sum_{p\in P_{ij}^\omega} (e_i + t_{ij}^p - e_j)^+ y_{ij}^{p\omega}, \quad \omega \in \Omega, j \in N \end{aligned} \tag{1g}$$

$$\begin{aligned} a_i^\omega \leq & \sum_{j:(i,j)\in\delta^+(i)} \min\{l_i, l_j - t_{ij}\} x_{ij} - \sum_{j:(i,j)\in\delta^+(i), l_i \geq l_j - t_{ij}} \sum_{p\in P_{ij}^\omega} (t_{ij}^p - t_{ij}) y_{ij}^{p\omega} \\ & - \sum_{j:(i,j)\in\delta^+(i), l_i < l_j - t_{ij}} \sum_{p\in P_{ij}^\omega} (l_i - (l_j - t_{ij}^p))^+ y_{ij}^{p\omega}, \quad \omega \in \Omega, i \in N \end{aligned} \tag{1h}$$

$$x_{ij} \in \{0,1\}, \quad (i,j) \in A \tag{1i}$$

$$y_{ij}^{p\omega} \in \{0,1\}, \quad \omega \in \Omega, (i,j) \in A, p \in P_{ij}^\omega. \tag{1j}$$

The objective function aims to minimize the total cost, which comprises the routing cost for serving regular customers and the expected recourse cost across all scenarios, including the additional routing cost for serving stochastic customers and the outsourcing cost for unserved ones. Constraints (1b) and (1c) are degree constraints, which guarantee that each regular customer is visited exactly once. Constraints (1d) ensure that in each scenario ω each stochastic customer k can be served at most once. Constraints (1e) indicate that for each scenario ω and for each arc (i,j), at most one path can be inserted in between i and j if the arc is used. Constraints (1f) eliminate possible subtours and ensure that the arrival time at regular customer j is at least the arrival time at regular customer i plus the direct travel time from i to j, provided that j is scheduled to be visited immediately after i with no stochastic customer in between; and if a path p is inserted between i and j, then the arrival time at j must be at least the arrival time at i plus the travel time from i to j via p. Constraints (1g) and (1h) enforce the time window requirements of regular customers, and ensure that the vehicles return to the depot by time T. Let $j \in N$. For $i \in N_0 \setminus \{j\}$ such that $(i,j) \in \delta^-(j)$ and $x_{ij} = 1$, if there exists $p \in P_{ij}^\omega$ such that $y_{ij}^{p\omega} = 1$, then (1g) becomes $a_j^\omega \geq e_i + t_{ij}^p$ if $e_j \leq e_i + t_{ij}$, and $a_j^\omega \geq e_j + (e_i + t_{ij}^p - e_j)^+$ otherwise; and on the other hand, if $y_{ij}^{p\omega} = 0$ for all $p \in P_{ij}^\omega$, then (1g) reduces to $a_j^\omega \geq \max\{e_j, e_i + t_{ij}\}$. In both situations, the resulting constraint is satisfied by a feasible solution and implies the time window lower bound (LB) constraint $a_j^\omega \geq e_j$. An analogous reasoning based on (1h) also applies to the time window upper bound (UB) constraint $a_j^\omega \leq l_j$ and to the maximum route duration constraint $a_j^\omega \leq T$. Constraints (1i) and (1j) define the domains of variables.

3.2.2. Single-commodity flow-based formulation

We introduce an additional continuous variable b_{ij}^ω representing the arrival time at a regular customer $j \in N$ in scenario $\omega \in \Omega$ when $x_{ij} = 1$ for $(i,j) \in A$, and 0 otherwise. The SCFF is obtained by replacing constraints (1f) - (1h) in the MTZF (1) with (2a) - (2e):

$$\sum_{j:(i,j)\in\delta^+(i)} b_{ij}^\omega \geq \sum_{j:(j,i)\in\delta^-(i)} b_{ji}^\omega + \sum_{j:(i,j)\in\delta^+(i)} t_{ij}x_{ij} + \sum_{j:(i,j)\in\delta^+(i)} \sum_{p\in P_{ij}^\omega} (t_{ij}^p - t_{ij})y_{ij}^{p\omega},$$
$$\omega \in \Omega, i \in N \tag{2a}$$

$$b_{ij}^\omega \geq (e_i + t_{ij})x_{ij} + \sum_{p\in P_{ij}^\omega} (t_{ij}^p - t_{ij})y_{ij}^{p\omega}, \quad \omega \in \Omega, (i,j) \in A : e_j \leq e_i + t_{ij} \tag{2b}$$

$$b_{ij}^\omega \geq e_j x_{ij} + \sum_{p\in P_{ij}^\omega} (e_i + t_{ij}^p - e_j)^+ y_{ij}^{p\omega}, \quad \omega \in \Omega, (i,j) \in A : e_j > e_i + t_{ij} \tag{2c}$$

$$b_{ij}^\omega \leq l_j x_{ij}, \quad \omega \in \Omega, (i,j) \in A \tag{2d}$$

$$
\begin{aligned}
\sum_{j:(j,i)\in\delta^-(i)} b_{ji}^{\omega} \leq & \sum_{j:(i,j)\in\delta^+(i)} \min\{l_i, l_j - t_{ij}\} x_{ij} \\
& - \sum_{j:(i,j)\in\delta^+(i), l_i \geq l_j - t_{ij}} \sum_{p\in P_{ij}^{\omega}} (t_{ij}^p - t_{ij}) y_{ij}^{p\omega} \\
& - \sum_{j:(i,j)\in\delta^+(i), l_i < l_j - t_{ij}} \sum_{p\in P_{ij}^{\omega}} (l_i - (l_j - t_{ij}^p))^+ y_{ij}^{p\omega}, \quad \omega\in\Omega, i\in N. \qquad \text{(2e)}
\end{aligned}
$$

Constraints (2a) serve to eliminate potential subtours and to properly link the arrival times between regular customers, accounting for the possible insertion of stochastic customers along the route. Specifically, if regular customer j is visited immediately after regular customer i with no stochastic customer inserted in between, the constraint ensures that the arrival time at j is at least the arrival time at i plus the direct travel time from i to j. Conversely, if a path p is inserted between i and j, the constraint requires that the arrival time at j be no less than the arrival time at i plus the travel times from i to j via p. Constraints (2b) and (2c) enforce the LBs of the time windows for regular customers. Constraints (2d) impose the UBs of the time windows and ensure that all vehicles return to the depot no later than time T. Additionally, constraints (2e) are valid inequalities providing valid UBs on the arrival time at each regular customer i under each scenario ω. Specifically, consider the case where no stochastic customer is inserted between i and its successor j, i.e., $y_{ij}^{p\omega} = 0$ for all $p \in P_{ij}^{\omega}$. In this case, the latest feasible arrival at i is bounded by the smaller of l_i (the latest time allowed to start service at i) and $l_j - t_{ij}$ (the latest time at i so that j can still be visited within its time window after traveling from i to j). Now, suppose a path p is inserted between i and j. In this case, the feasible latest arrival time at i is constrained by both l_i and $l_j - t_{ij}^p$ (the latest arrival time at i ensuring that j can still be served within its time window via p). As such, when $l_i \geq l_j - t_{ij}$, since inserting p increases the extra travel time between i and j by $t_{ij}^p - t_{ij}$, the UB on the latest arrival time at i can be tightened by subtracting such extra time and it becomes $l_j - t_{ij}^p$. When $l_i < l_j - t_{ij}$, i's own time window is binding. If inserting p causes the effective bound $l_j - t_{ij}^p$ to drop below l_i, the UB on the latest arrival time at i can be reduced by $l_i - (l_j - t_{ij}^p)$; otherwise it stays at l_i.

3.3. DEF for VRPTW-SC-Skip

Different from VRPTW-SC-Insert, in VRPTW-SC-Skip, stochastic customers whose visit requests will be outsourced are decided in the first stage together with vehicle routes to serve all regular customers and the remaining stochastic customers. Each of these customers is visited exactly once within their time windows and all vehicles return to the depot by time T. In the second stage, in each scenario, routes are adapted by skipping the stochastic customers that do not request visits in that scenario. The objective is to minimize the total expected cost.

Next, we formulate VRPTW-SC-Skip. As the preliminary computational results for VRPTW-SC-Insert showed that SCFF performed better than MTZF, for VRPTW-SC-Skip, we only provide a flow formulation. However an MTZF can be derived similarly.

We introduce the following notation and decision variables. Let $p' = (k_1, k_2, \ldots, k_m)$ denote a path consisting of a subset of stochastic customers. Let P'_{ij} represent the set of all time- and cost-feasible paths that can be traversed in going from node i to node j such that arc $(i, j) \in A$. For each scenario $\omega \in \Omega$, let $c_{ij}^{p'\omega}$ and $t_{ij}^{p'\omega}$ denote the cost and time, respectively, associated with traveling from regular customer i to j via $p' \in P'_{ij}$, after skipping stochastic customers that are not realized in scenario ω. We introduce a binary decision variable $y_{ij}^{p'}$, which equals 1 if path $p' \in P'_{ij}$ is planned to be traversed in stage 1 while going from i to j with $(i, j) \in A$ when $x_{ij} = 1$, and 0 otherwise. To relate the two problems, we also say that path p' is inserted into arc (i, j) if $y_{ij}^{p'}$ is 1.

The SCFF for VRPTW-SC-Skip is as follows:

$$\min \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{(i,j)\in A} \sum_{p'\in P'_{ij}} \left(\sum_{\omega\in\Omega} \rho_\omega c_{ij}^{p'\omega} - c_{ij} \right) y_{ij}^{p'} + \sum_{k\in \bigcup_{\omega\in\Omega} N^\omega} \beta_k \Big(1 - \sum_{(i,j)\in A} \sum_{p'\in P'_{ij}:k\in p'} y_{ij}^{p'}\Big) \tag{3a}$$

subject to:

(1b), (1c), (1i), (2d)

$$\sum_{(i,j)\in A} \sum_{p'\in P'_{ij}:k\in p'} y_{ij}^{p'} \le 1, \quad k \in \bigcup_{\omega\in\Omega} N^\omega \tag{3b}$$

$$\sum_{p'\in P'_{ij}} y_{ij}^{p'} \le x_{ij}, \quad (i,j) \in A \tag{3c}$$

$$\sum_{j:(i,j)\in\delta^+(i)} b_{ij}^\omega \ge \sum_{j:(j,i)\in\delta^-(i)} b_{ji}^\omega + \sum_{j:(i,j)\in\delta^+(i)} t_{ij}x_{ij} + \sum_{j:(i,j)\in\delta^+(i)} \sum_{p'\in P'_{ij}} (t_{ij}^{p'\omega} - t_{ij}) y_{ij}^{p'}, \quad i \in N, \omega \in \Omega \tag{3d}$$

$$b_{ij}^\omega \ge (e_i + t_{ij})x_{ij} + \sum_{p'\in P'_{ij}} (t_{ij}^{p'\omega} - t_{ij}) y_{ij}^{p'}, \quad (i,j) \in A : e_j \le e_i + t_{ij}, \omega \in \Omega \tag{3e}$$

$$b_{ij}^\omega \ge e_j x_{ij} + \sum_{p'\in P'_{ij}} (e_i + t_{ij}^{p'\omega} - e_j)^+ y_{ij}^{p'}, \quad (i,j) \in A : e_j > e_i + t_{ij}, \omega \in \Omega \tag{3f}$$

$$
\begin{aligned}
\sum_{j:(j,i)\in\delta^-(i)} b_{ji}^{\omega} \leq \sum_{j:(i,j)\in\delta^+(i)} & \min\{l_i, l_j - t_{ij}\} x_{ij} \\
& - \sum_{j:(i,j)\in\delta^+(i), l_i \geq l_j - t_{ij}} \sum_{p'\in P'_{ij}} (t_{ij}^{p'\omega} - t_{ij}) y_{ij}^{p'} \\
& - \sum_{j:(i,j)\in\delta^+(i), l_i < l_j - t_{ij}} \sum_{p'\in P'_{ij}} (l_i - (l_j - t_{ij}^{p'\omega}))^+ y_{ij}^{p'}, \quad i \in N, \omega \in \Omega
\end{aligned} \tag{3g}
$$

$$
y_{ij}^{p'} \in \{0,1\}, \quad (i,j) \in A, p' \in P'_{ij}. \tag{3h}
$$

The objective function (3a) minimizes the expected routing and outsourcing costs. Constraints (3b) ensure that each stochastic customer is served at most once. Constraints (3c) enforce that, for each arc (i, j), at most one insertion path can be selected whenever the arc is used. The remaining constraints, (3d) - (3g), have the same interpretation as constraints (2a) - (2c) and (2e) from Section 3.2.2. Finally, constraints (3h) impose binary restrictions on the variables.

3.4. Comparison of the two a priori approaches

The key distinction between the two approaches lies in the flexibility of recourse actions. In VRPTW-SC-Insert, the routing of stochastic customers is deferred entirely to the second stage. For each scenario ω, the decision maker observes the realized set of stochastic customers N^ω and then decides, adaptively, which customers to insert into the planned routes and which to outsource.

In contrast, VRPTW-SC-Skip commits to a fixed insertion plan for all stochastic customers before observing their realization. For each arc (i, j) used in the first-stage routes, if a path p' composed of potential stochastic customers is preselected, then in scenario ω, the vehicle must traverse this path, skipping those that did not materialize. This means that in this approach, the recourse action is predetermined and cannot adapt to the actual set of realized customers.

Proposition 1. *Let z^*_{insert} denote the optimal value of the VRPTW-SC-Insert and let z^*_{skip} denote the optimal value of the VRPTW-SC-Skip. Then,*

$$
z^*_{insert} \leq z^*_{skip}.
$$

Proof. Consider any feasible solution to VRPTW-SC-Skip. We construct a feasible solution to VRPTW-SC-Insert as follows: (i) we keep the same first-stage routing decisions

x, and (ii) for each scenario ω and each arc (i,j) with $x_{ij} = 1$, if $y_{ij}^{p'} = 1$ then we insert into arc (i,j) only those stochastic customers in p' that are realized in the set N^ω, preserving their order in p'. This constructed solution is feasible for VRPTW-SC-Insert. In every scenario, the cost incurred by this constructed solution is equal to that incurred by VRPTW-SC-Skip. Hence z^*_{insert} cannot be larger than z^*_{skip}. □

3.5. Valid inequalities

In this section, we present valid inequalities that can be used to strengthen the DEFs for both the VRPTW-SC-Insert and VRPTW-SC-Skip.

Short Infeasible Path Constraint (SIPC): Infeasible path constraints are introduced by Ascheuer et al. (2000) to model the asymmetric TSP with time windows. Here we present simple lifted SIPCs. For $\omega \in \Omega$, $(i_1, i_2) \in A$ with $i_1, i_2 \in N$, let

$$F_{i_1 i_2} = \{i_3 \in N_0 : (i_2, i_3) \in A, (e_{i_1} + t_{i_1 i_2} + t_{i_2 i_3} > l_{i_3} \text{ or } i_3 = i_1)\},$$

$$B_{i_1 i_2} = \{i_3 \in N_0 : (i_3, i_1) \in A, (e_{i_3} + t_{i_3 i_1} + t_{i_1 i_2} > l_{i_2} \text{ or } i_3 = i_2)\}.$$

Sets F_{i_1,i_2} and B_{i_1,i_2} denote the sets of nodes (potentially including both regular customers and the depot) that cannot come after and before arc (i_1, i_2), respectively, in any feasible route. Hence the inequalities

$$x_{i_1 i_2} + \sum_{i_3 \in F_{i_1 i_2}} x_{i_2 i_3} \leq 1, \tag{4}$$

$$\sum_{i_3 \in B_{i_1 i_2}} x_{i_3 i_1} + x_{i_1 i_2} \leq 1 \tag{5}$$

are valid.

Let $i_1, i_3 \in N$ with $i_1 < i_3$, $i_2 \in N$ be distinct nodes such that (i_1, i_2), (i_2, i_1), (i_2, i_3), $(i_3,\, i_2) \in A$, $e_{i_1} + t_{i_1 i_2} + t_{i_2 i_3} > l_{i_3}$ and $e_{i_3} + t_{i_3 i_2} + t_{i_2 i_1} > l_{i_1}$. The inequality

$$x_{i_1 i_2} + x_{i_2 i_1} + x_{i_2 i_3} + x_{i_3 i_2} \leq 1 \tag{6}$$

is valid.

Let $i_2 \in N$, $N_1, N_3 \subset N_0$ be such that $(i_1, i_2), (i_2, i_3) \in A$ and $e_{i1} + t_{i_1 i_2} + t_{i_2 i_3} > l_{i_3}$ for all $i_1 \in N_1$ and $i_3 \in N_3$. The inequality

$$\sum_{i_1 \in N_1} x_{i_1 i_2} + \sum_{i_3 \in N_3} x_{i_2 i_3} \leq 1 \tag{7}$$

is valid.

Subtour Elimination Constraint (SEC)(Bard et al., 2002): Let $S \subseteq N$ with $|S| \geq 2$. The SEC

$$x(A(S)) \leq |S| - \lambda_S, \tag{8}$$

is valid, where λ_S denotes the number of vehicles required to serve all nodes in S. The SEC can be equivalently reformulated as the following connectivity constraint

$$x(\delta^+(S)) \geq \lambda_S, \tag{9}$$

which ensures that at least λ_S arcs leave S, thereby maintaining the necessary level of connectivity to avoid subtours.

4. Solution methods

Due to Proposition 1, we know that the optimal value of VRPTW-SC-Insert is not more than the one of VRPTW-SC-Skip, so in this section, we present two exact algorithms to solve VRPTW-SC-Insert: a C&B algorithm that incorporates the separation of connectivity constraints (9), and an ILS algorithm tailored to exploit the problem's two-stage stochastic structure. Similar algorithms can also be developed to solve VRPTW-SC-Skip. In our computational results, we adapt our C&B algorithm for this purpose.

4.1. Preprocessing (Time window reduction)

We reduce the width of the time windows for regular customers by iteratively adjusting the time windows for each $k \in N$ until no further tightening can be made (Ascheuer et al., 2001):

Step 1. $e_k \leftarrow \max\{e_k, \min_{i:(i,k)\in\delta^-(k)} \{e_i + t_{ik}\}\}$
Step 2. $l_k \leftarrow \min\{l_k, \max_{j:(k,j)\in\delta^+(k)} \{l_j - t_{kj}\}\}$

4.2. Cut-and-branch algorithm

In this subsection, we describe our C&B algorithm, where the separation procedure for the connectivity inequality (9) follows a routine similar to that proposed by Ozbaygin et al. (2016). The algorithmic framework consists of the following main steps:

i) Initialization: We initialize the strengthened formulation for the VRPTW-SC-Insert by incorporating the SIPCs (4) - (6), along with a subset of SIPC (7) generated by the routine introduced in Algorithm 1.

ii) Separation at the root node: We construct the support graph $\bar{G} = (N_0, E)$ induced by the fractional solution $\bar{\mathbf{x}}$ to the LP-relaxed strengthened formulation, where the edge set is defined as $E = \big\{\{i,j\} : \bar{x}_{ij} > 0, (i,j) \in A \text{ or } \bar{x}_{ji} > 0, (j,i) \in A\big\}$. If $\bar{G}$ is disconnected, then for each connected component $\mathcal{S}$ not containing the depot, we compute the minimum number of vehicles required, $\lambda_{\mathcal{S}}$, to serve the component. We then identify and add the valid connectivity cut (9). If $\bar{G}$ is connected, we assign a capacity to the edge $\{i,j\}$ equal to $\bar{x}_{ij} + \bar{x}_{ji}$, and then compute the global minimum cut. Let $\mathcal{S}$ denote the resulting node set without depot, and $Cap(\mathcal{S})$ denote the cut value. If $\frac{1}{2}Cap(\mathcal{S}) < \lambda_{\mathcal{S}}$, we again add the valid connectivity cut as above.

iii) B&B and termination: After processing the root node, the remaining B&B procedure is handled by Gurobi's built-in mixed-integer programming (MIP) solver. The algorithm terminates upon finding an optimal solution or reaching the time limit.

Algorithm 1: Separation of SIPC (7)

Input: Graph $G = (N_0, A)$, time windows $[e_i, l_i]$, travel times t_{ij}

Output: A subset of SIPC (7)

foreach $i_2 \in N$ **do**
    foreach $i_1 \in N_0 \setminus \{i_2\}$ *such that* $(i_1, i_2) \in A$ **do**
        $N_3(i_1) \leftarrow \{\, i_3 \in N_0 \setminus \{i_1, i_2\} \mid (i_2, i_3) \in A,\ e_{i_1} + t_{i_1 i_2} + t_{i_2 i_3} > l_{i_3} \,\}$
    foreach $i_1 \in N_0 \setminus \{i_2\}$ **do**
        if $N_3(i_1) \neq \emptyset$ *and if it has not already been considered* **then**
            $N_1 \leftarrow \{\, i_1' \in N_0 \mid N_3(i_1) \subseteq N_3(i_1') \,\}$, $N_3 \leftarrow N_3(i_1)$
            Add a valid inequality (7) for (N_1, N_3)

4.3. Integer L-shaped algorithm

In this subsection, we detail an ILS algorithm developed based on Angulo et al. (2016), which improves the original method of Laporte and Louveaux (1993) by using the LP-relaxed SPs to generate Benders cuts whenever such cuts are sufficient to eliminate sub-optimal solutions, to address the VRPTW-SC-Insert. We first describe the decomposition scheme based on Benders decomposition (BD) (Benders, 1962), followed by several algorithmic enhancements incorporated to improve the algorithm's performance.

4.3.1. Benders reformulations

Benders reformulation of MTZF (1)

We introduce an auxiliary decision variable θ^ω that approximates the second-stage cost for scenario ω and formulate the MP as a VRPTW for regular customers by decomposing MTZF (1) as follows:

$$\min \sum_{(i,j)\in A} c_{ij}x_{ij} - \sum_{\omega\in\Omega} \rho_\omega \theta^\omega + \sum_{\omega\in\Omega} \rho_\omega \sum_{k\in N^\omega} \beta_k \tag{10a}$$

subject to:

(1*b*), (1*c*), (1*i*)

$$a_j \geq a_i + t_{ij}x_{ij} - (l_i - e_j)(1 - x_{ij}), \quad (i,j) \in A : i, j \neq 0 \tag{10b}$$

$$a_j \geq \sum_{i:(i,j)\in\delta^-(j)} \max\{e_j, e_i + t_{ij}\}x_{ij}, \quad j \in N \tag{10c}$$

$$a_i \leq \sum_{j:(i,j)\in\delta^+(i)} \min\{l_i, l_j - t_{ij}\}x_{ij}, \quad i \in N \tag{10d}$$

$$\text{Benders cuts} \tag{10e}$$

$$\text{Optimality cuts,} \tag{10f}$$

where the objective (10a) minimizes the total cost composed of the routing cost for regular customers and the expected recourse cost across all scenarios. To ensure that all SPs remain feasible for any candidate first-stage decision, we augment the MP with additional constraints (10b) - (10d), thereby achieving a complete recourse. These constraints, meanwhile, prevent subtours and enforce both time windows of regular customers and the maximum route duration. Constraint (10e) represents Benders cuts generated iteratively from the LP-relaxed SPs, whereas constraint (10f) denotes optimality cuts derived from the exact solutions of SPs, which are progressively added to ensure convergence of the algorithm to an optimal solution.

After solving the MP, we obtain its optimal solution, denoted by the tuple $(\bar{\mathbf{x}}, \bar{\theta})$. Given a fixed first-stage decision $\bar{\mathbf{x}}$, the MTZ-based SP associated with scenario $\omega \in \Omega$ is stated as

$$\theta^\omega(\bar{\mathbf{x}}) = \max \sum_{(i,j)\in A} \sum_{p\in P_{ij}^\omega} (c_{ij} + \sum_{k\in p} \beta_k - c_{ij}^p) y_{ij}^p \tag{11a}$$

subject to:

$$\sum_{(i,j)\in A} \sum_{p\in P_{ij}^\omega : k\in p} y_{ij}^p \leq 1, \quad k \in N^\omega \tag{11b}$$

$$\sum_{p\in P_{ij}^{\omega}} y_{ij}^p \le \bar{x}_{ij}, \quad (i,j)\in A \tag{11c}$$

$$a_j \ge a_i + t_{ij}\bar{x}_{ij} + \sum_{p\in P_{ij}^{\omega}} (t_{ij}^p - t_{ij})y_{ij}^p - (l_i - e_j)(1-\bar{x}_{ij}), \quad (i,j)\in A : i,j\neq 0 \tag{11d}$$

$$\begin{aligned} a_j \ge & \sum_{i:(i,j)\in\delta^-(j)} \max\{e_j, e_i + t_{ij}\}\bar{x}_{ij} + \sum_{i:(i,j)\in\delta^-(j), e_j\le e_i+t_{ij}} \sum_{p\in P_{ij}^{\omega}} (t_{ij}^p - t_{ij})y_{ij}^p \\ & + \sum_{i:(i,j)\in\delta^-(j), e_j> e_i+t_{ij}} \sum_{p\in P_{ij}^{\omega}} (e_i + t_{ij}^p - e_j)^+ y_{ij}^p, \quad j\in N \end{aligned} \tag{11e}$$

$$\begin{aligned} a_i \le & \sum_{j:(i,j)\in\delta^+(i)} \min\{l_i, l_j - t_{ij}\}\bar{x}_{ij} - \sum_{j:(i,j)\in\delta^+(i), l_i\ge l_j-t_{ij}} \sum_{p\in P_{ij}^{\omega}} (t_{ij}^p - t_{ij})y_{ij}^p \\ & - \sum_{j:(i,j)\in\delta^+(i), l_i< l_j-t_{ij}} \sum_{p\in P_{ij}^{\omega}} (l_i - (l_j - t_{ij}^p))^+ y_{ij}^p, \quad i\in N \end{aligned} \tag{11f}$$

$$y_{ij}^p \in \{0,1\}, \quad (i,j)\in A, p\in P_{ij}^{\omega}. \tag{11g}$$

For each SP, all the variables and constraints maintain the same interpretation as in Section 3.2.1 but dropping the scenario index ω.

Since the MP (10) is a relaxation of (1), $\bar{\theta}^{\omega}$ obtained by solving the MP is an overestimate of the objective value for the SP with respect to the fixed first-stage decision $\bar{\mathbf{x}}$. As such, when $\bar{\theta}^{\omega}$ is greater than the optimal value of the LP-relaxed SP under scenario ω with $\bar{\mathbf{x}}$ passed from the optimally solved MP, denoted by $\theta_{LP}^{\omega}(\bar{\mathbf{x}})$, i.e., $\bar{\theta}^{\omega} > \theta_{LP}^{\omega}(\bar{\mathbf{x}})$, we add the following Benders cut to the MP to remove the current solution:

$$\begin{aligned} \theta^{\omega} \le & \sum_{k\in N^{\omega}} \bar{\alpha}_k^{'} + \sum_{(i,j)\in A} \bar{\gamma}_{ij}^{'} x_{ij} - \sum_{(i,j)\in A: i,j\neq 0} (t_{ij}x_{ij} - (l_i - e_j)(1-x_{ij}))\bar{\zeta}_{ij}^{'} \\ & - \sum_{j\in N} \sum_{i:(i,j)\in\delta^-(j)} \max\{e_j, e_i + t_{ij}\}\bar{\eta}_j^{'} x_{ij} + \sum_{i\in N} \sum_{j:(i,j)\in\delta^+(i)} \min\{l_i, l_j - t_{ij}\}\bar{\lambda}_i^{'} x_{ij}, \end{aligned} \tag{12}$$

where $(\bar{\alpha}^{'}, \bar{\gamma}^{'}, \bar{\zeta}^{'}, \bar{\eta}^{'}, \bar{\lambda}^{'})$ is an optimal solution to the dual MTZ-based LP-relaxed SP under scenario ω.

On the other hand, when $\bar{\theta}^{\omega} \le \theta_{LP}^{\omega}(\bar{\mathbf{x}})$, if the optimal solution to the LP-relaxed scenario SP is integer, we accept the solution; and otherwise, we solve the mixed-integer scenario SP to obtain its optimal value, denoted by $\theta_{MIP}^{\omega}(\bar{\mathbf{x}})$. When $\bar{\theta}^{\omega} > \theta_{MIP}^{\omega}(\bar{\mathbf{x}})$, inspired by Laporte and Louveaux (1993), we incorporate the following adapted ILS optimality cut, tailored to the problem structure, to the MP to remove the solution:

$$\theta^{\omega} \le \theta_{MIP}^{\omega}(\bar{\mathbf{x}}) + (\mathrm{UB}^{\omega} - \theta_{MIP}^{\omega}(\bar{\mathbf{x}})) \sum_{(i,j)\in A: \bar{x}_{ij}=0} x_{ij}, \tag{13}$$

where $\mathrm{UB}^\omega = \sum_{k \in N^\omega} \beta_k$ for $\omega \in \Omega$, which is also used as a primal UB for the auxiliary variable θ^ω in the MP. Note that the optimality cut does not include terms for arcs $(i', j') \in A$ with $\bar{x}_{i'j'} = 1$ since if some arc (i', j') with $\bar{x}_{i'j'} = 1$ takes value 0, then another arc (i, j) with $\bar{x}_{ij} = 0$ must take value 1.

Next, we reformulate SCFF using BD in the same way as we did for MTZF.

Benders reformulation of SCFF (2)

The MP for the decomposition of SCFF (2) is formulated by replacing constraints (10b) - (10d) in the model (10) with the constraints (14a) - (14d):

$$\sum_{j:(i,j)\in\delta^+(i)} b_{ij} \geq \sum_{j:(j,i)\in\delta^-(i)} b_{ji} + \sum_{j:(i,j)\in\delta^+(i)} t_{ij}x_{ij}, \quad i \in N \tag{14a}$$

$$b_{ij} \geq \max\{e_j, e_i + t_{ij}\}x_{ij}, \quad (i,j) \in A \tag{14b}$$

$$b_{ij} \leq l_j x_{ij}, \quad (i,j) \in A \tag{14c}$$

$$\sum_{j:(j,i)\in\delta^-(i)} b_{ji} \leq \sum_{j:(i,j)\in\delta^+(i)} \min\{l_i, l_j - t_{ij}\}x_{ij}, \quad i \in N. \tag{14d}$$

Constraints (14a) - (14d) eliminate subtours and enforce compliance with both time window requirements of regular customers and the maximum route duration. By incorporating these constraints, the single-commodity flow (SCF)-based MP guarantees complete recourse feasibility.

For a given MP solution $\bar{\mathbf{x}}$, the SCF-based SP under each scenario $\omega \in \Omega$ is formulated by replacing constraints (11d) - (11f) in the model (11) with the constraints (15a) - (15e):

$$\sum_{j:(i,j)\in\delta^+(i)} b_{ij} \geq \sum_{j:(j,i)\in\delta^-(i)} b_{ji} + \sum_{j:(i,j)\in\delta^+(i)} t_{ij}\bar{x}_{ij} + \sum_{j:(i,j)\in\delta^+(i)} \sum_{p\in P^\omega_{ij}} (t^p_{ij} - t_{ij})y^p_{ij},$$
$$i \in N \tag{15a}$$

$$b_{ij} \geq (e_i + t_{ij})\bar{x}_{ij} + \sum_{p\in P^\omega_{ij}} (t^p_{ij} - t_{ij})y^p_{ij}, \quad (i,j) \in A : e_j \leq e_i + t_{ij} \tag{15b}$$

$$b_{ij} \geq e_j\bar{x}_{ij} + \sum_{p\in P^\omega_{ij}} (e_i + t^p_{ij} - e_j)^+ y^p_{ij}, \quad (i,j) \in A : e_j > e_i + t_{ij} \tag{15c}$$

$$b_{ij} \leq l_j\bar{x}_{ij}, \quad (i,j) \in A \tag{15d}$$

$$\sum_{j:(j,i)\in\delta^-(i)} b_{ji} \leq \sum_{j:(i,j)\in\delta^+(i)} \min\{l_i, l_j - t_{ij}\}\bar{x}_{ij}$$
$$- \sum_{j:(i,j)\in\delta^+(i), l_i \geq l_j - t_{ij}} \sum_{p\in P^\omega_{ij}} (t^p_{ij} - t_{ij})y^p_{ij}$$

$$-\sum_{j:(i,j)\in\delta^+(i),l_i<l_j-t_{ij}}\sum_{p\in P_{ij}^{\omega}}(l_i-(l_j-t_{ij}^p))^+y_{ij}^p, \quad i\in N. \tag{15e}$$

Likewise, for each SP, constraints (15a)–(15e) retain the same meaning as in Section 3.2.2, with the scenario index ω being removed.

The corresponding Benders cut is as follows:

$$\begin{aligned}\theta^{\omega}\leq &\sum_{k\in N^{\omega}}\bar{\alpha}_k''+\sum_{(i,j)\in A}\bar{\gamma}_{ij}''x_{ij}-\sum_{i\in N}\sum_{j:(i,j)\in\delta^+(i)}t_{ij}\bar{\eta}_i''x_{ij}-\sum_{(i,j)\in A}\max\{e_j,e_i+t_{ij}\}\bar{\xi}_{ij}''x_{ij}\\&+\sum_{(i,j)\in A}l_j\bar{\zeta}_{ij}''x_{ij}+\sum_{i\in N}\sum_{j:(i,j)\in\delta^+(i)}\min\{l_i,l_j-t_{ij}\}\bar{\lambda}_i''x_{ij},\end{aligned}\tag{16}$$

where $(\bar{\alpha}'',\bar{\gamma}'',\bar{\eta}'',\bar{\xi}'',\bar{\zeta}'',\bar{\lambda}'')$ is an optimal solution to the dual SCF-based LP-relaxed SP under scenario ω. Furthermore, the adapted ILS optimality cut (13) is employed to guarantee the algorithmic convergence.

4.3.2. Algorithmic enhancements

We implement several algorithmic enhancements to improve the computational performance of our ILS algorithm.

B&C implementation: We solve the Benders reformulations for the VRPTW-SC-Insert within a B&C framework, implemented on a single B&B tree for the MP.

Warm start strategies: First, we initialize the LB of the problem by solving the LP-relaxed MP and progressively adding Benders cuts derived from the dual solutions of the LP-relaxed SPs. This iterative process continues until no further violated Benders cuts are identified, or until the number of added Benders cuts exceeds 500, thereby resulting in a stronger LP relaxation. Second, we initialize the UB of the problem by injecting a feasible solution $(\bar{\mathbf{x}},\bar{\theta})$ to the MP, where all auxiliary variables $\bar{\theta}^{\omega}$ are set to zero. The feasible solution $\bar{\mathbf{x}}$ is obtained by solving a modified MP within a given time limit, where the objective minimizes the total routing cost of serving regular customers, and Benders and optimality cuts are excluded. Third, we strengthen the MP by incorporating the lifted SIPCs (4)-(7).

Benders cuts and connectivity cuts (9) at the root node: At the root node of the B&B tree, we identify and add two classes of cuts: Benders cuts, derived from the dual solutions of the LP-relaxed SPs, to improve the LB, and valid connectivity cuts, generated for fractional solutions, to eliminate infeasible subtours and to enforce vehicle connectivity requirements.

Heuristic solutions in B&C: At each integral node of the B&B tree, a feasible first-stage solution can be combined with scenario-specific second-stage solutions to construct a feasible solution to the original two-stage stochastic problem. Specifically, consider a feasible first-stage solution $(\bar{\mathbf{x}}, \bar{\theta})$, where $\bar{\mathbf{x}}$ is integral. For each scenario $\omega \in \Omega$, we proceed as follows: if solving the LP-relaxed SP yields a feasible solution with integral values for $\bar{\mathbf{y}}$, then the associated optimal value $\theta^{\omega}_{LP}(\bar{\mathbf{x}})$, also denoted as $\bar{\theta}^{\omega}(\bar{\mathbf{x}})$, is collected directly; and otherwise, if the auxiliary variable satisfies $\bar{\theta}^{\omega} \leq \theta^{\omega}_{LP}(\bar{\mathbf{x}})$, we collect the SP optimal value $\bar{\theta}^{\omega}(\bar{\mathbf{x}})$ from solving the SP as a MIP. Once optimal values $\bar{\theta}^{\omega}(\bar{\mathbf{x}})$ are collected for all scenarios, we compute the objective function value of the MP at the candidate solution $(\bar{\mathbf{x}}, \bar{\theta}(\bar{\mathbf{x}}))$. If this solution improves upon the current best known UB, we update the UB accordingly via Gurobi's callback.

Non-dominated Benders cuts: To generate strong Benders cuts that are not dominated by others, we adopt the approach proposed by Sherali and Lunday (2013). Specifically, we initialize the core point with $\mathbf{x^0} = \mathbf{1}$, and at each iteration t, update it using the rule $x^{0,t}_{ij} = \frac{1}{2}x^{0,t-1}_{ij} + \frac{1}{2}\bar{x}^{t}_{ij}$, where $\bar{x}^t$ denotes the current first-stage solution. A small perturbation parameter is set as $\epsilon = 10^{-12}$.

Multicut strategy: We adopt a multicut strategy in which individual Benders and optimality cuts are generated and added to the MP separately for each scenario-specific variable θ^{ω}.

5. Computational experiments

In this section, we introduce the instance generation and implementation details of the proposed solution methods. We present a comprehensive analysis of the computational results for VRPTW-SC-Insert, focusing on: the effectiveness of incorporating VIs in strengthening the formulations, a performance comparison of solution approaches based on MTZF and SCFF, and the impact of varying insertion path length settings on solution quality. We also provide a comparison of the optimal values of VRPTW-SC-Insert and VRPTW-SC-Skip.

5.1. Experimental setup

5.1.1. Instance generation

We generate test instances based on the Solomon benchmark sets C1, R1, and RC1 (Solomon, 1987), which feature clustered, random, and mixed distributions of customers, respectively. To ensure that all selected regular customers possess time windows, we choose the following instances as the basis for our test set: C101, 105–109, R101, 105, 109–112, and RC101, 105–108.

The configuration of each test instance is as follows. The first node of the benchmark instance is designated as the depot, and regular customers $|N| = 25$ are randomly selected nodes from the instance, and their original time windows are retained. The maximum return time to the depot is set equal to the workday duration T. Travel times t_{ij} and travel costs c_{ij} are both initially set to the integer-rounded Euclidean distance from nodes i to j. To guarantee the triangle inequality, these distances are updated based on the shortest paths using the Floyd-Warshall algorithm. The service time at node i is subsequently incorporated by adding it to the travel time t_{ij} for all outgoing arcs (i, j).

We consider a fixed pool of 10 stochastic customers, each randomly selected from the nodes in the benchmark instance (excluding those designated as the depot and regular customers). The availability of each stochastic customer is modeled as an independent Bernoulli random variable with presence probability $p_s = 0.5$. Scenarios $\omega \in \Omega$, each characterized by a realized set of stochastic customers N^ω, are generated by sampling without replacement until the desired number of scenarios is reached. All scenarios are assigned equal probability weights, $\rho_\omega = \frac{1}{|\Omega|}$. We consider multiple scenario set sizes, with $|\Omega| \in \{10, 30, 50, 100\}$. For each stochastic customer $k \in N^\omega$, a fixed outsourcing cost is 100.

5.1.2. Implementation details

All computational experiments are conducted on a 64-bit Windows computer equipped with a 3.20 GHz AMD Ryzen 7 6800H processor and 16 GB memory running. All solution methods are implemented in C++ using Gurobi 12.0.2 as the LP and MIP solver. Gurobi is configured to use up to 12 threads for all experiments. All other solver parameters, including *Presolve*, *Heuristics*, and *Cuts*, are kept at their default values unless explicitly noted. The computation time is measured in seconds, and a time limit of 3600 seconds (one hour) is imposed for each run, excluding warm-start procedures.

To determine the set of paths composed solely of stochastic customers that can be inserted into arc $(i, j) \in A$, i.e., P_{ij}^ω and P'_{ij}, we employ a Depth-First Search algorithm to identify all feasible paths. For P_{ij}^ω, among feasible paths consisting of the same set of stochastic customers but differing in sequence, we retain only the path with the minimum routing cost for traveling from i to j.

For the compact formulations solved by Gurobi, a warm-start strategy is implemented to ensure that a feasible solution is always available upon termination. This is achieved by injecting an initial solution to the solver, where x_{ij} variables are assigned values from a VRPTW solution for regular customers (obtained under a 60-second time limit with the objective of minimizing routing cost), and all $y_{ij}^{p\omega}$ variables are set to zero, thereby providing a valid starting point for the optimization. The same warm-start strategy is consistently applied in both the C&B and the ILS algorithms for VRPTW-SC-Insert. The value of λ_S required in the connectivity inequality (9) is obtained by solving a VRPTW (with the objective of minimizing the number of vehicles within a 10-second time limit). If solved to optimality, λ_S is set to the optimal value; otherwise, it is estimated by taking the ceiling of the best LB at termination. To support separation procedures, we leverage Boost C++ Libraries (*boost.org*) for two key components: the connected-components algorithm is used to detect graph connectivity, and the Stoer-Wagner algorithm is used to compute a global minimum cut. Moreover, Gurobi's generic callback is utilized to dynamically generate and add valid connectivity cuts, Benders cuts and optimality cuts during the solution process.

5.2. Experimental results

5.2.1. Effects of VIs for VRPTW-SC-Insert

To evaluate the efficacy of the VIs from Section 3.5, we incrementally add them to both MTZF (1) and SCFF (2) for VRPTW-SC-Insert, measuring improvements in root node LBs. To isolate the impact of these inequalities, we herein disable Gurobi's *Presolve*, *Heuristics*, and *Cuts*, ensuring LB improvements arise solely from the added VIs.

We begin by establishing baseline values by computing the LP relaxation bounds of both MTZF (1) and SCFF (2), denoted as LB_r^o, on instances with $|\Omega| = 10$. We subsequently strengthen these formulations by progressively incorporating the VIs: the SIPCs (4)-(7) first and then the connectivity constraint (9). The resulting improved root node LB after strengthening is denoted as LB_r^+. The percentage improvement at the root node is computed as $\frac{LB_r^+ - LB_r^o}{LB_r^o} \times 100\%$.

Table 1 presents the average percentage improvements in root node LBs obtained by applying different classes of VIs to both MTZF (1) and SCFF (2), with results organized by instance set. In the table, the column '*Instance set*' identifies each instance set, with the number of instances in parentheses. The column 'z^*' presents the average optimal values. The columns 'LB_r^o' report the average LP relaxation bounds for both formulations, respectively. The columns '*+(4)-(7)*' show the average percentage improvements in the root node LBs after adding the SIPCs (4)-(7) as opposed to their LP relaxation bounds. The columns '*+(4)-(7)+(9)*' present the average percentage improvements when combining the SIPCs (4)-(7) with the connectivity constraints (9).

Table 1: Effects of VIs on the root node LB

		MTZF (1)			SCFF (2)		
Instance set	z^*	LB_r^o	+(4)-(7)	+(4)-(7)+(9)	LB_r^o	+(4)-(7)	+(4)-(7)+(9)
C (30)	540.01	499.55	7.67%	7.89%	505.41	6.35%	6.46%
R (30)	506.12	412.66	7.87%	11.73%	458.65	4.28%	4.81%
RC (25)	611.30	463.58	12.28%	18.42%	529.21	7.15%	8.37%

Table 1 demonstrates the incremental improvements in the root node LBs achieved by incorporating two classes of VIs, i.e., SIPCs (4)-(7) and the connectivity constraint (9), into both MTZF (1) and SCFF (2). Additionally, it is also observed that SCFF consistently exhibits stronger average baseline LBs (LB_r^o) than MTZF across all instance sets, suggesting its inherent formulation tightness. Adding SIPCs (4)-(7) alone improves LBs on average by 4.28 - 12.28%, with MTZF benefiting more than SCFF. Further incorporating the connectivity constraint (9) yields cumulative average improvements of 4.81 - 18.42%, with MTZF again showing larger gains.

In summary, the VIs from Section 3.5 strengthen both formulations, leading to tighter root node LBs. Notably, their impact is more pronounced for the MTZF with an initially weaker LP relaxation.

5.2.2. Comparison of solution approaches based on MTZF and SCFF for VRPTW-SC-Insert

We enhance both formulations by incorporating the VIs from Section 3.5 and develop two exact solution algorithms: a C&B algorithm and an ILS algorithm. To evaluate the performance improvement of the strengthened formulations compared to their original counterparts, the relative effectiveness of C&B and ILS algorithms on the strengthened formulations, and the performance differences of solution approaches between MTZF and SCFF, we conduct computational experiments on instances with $|\Omega| = 10$.

Table 2 compares the computational performance of solution approaches applied to both formulations: Gurobi on the compact formulation, C&B and ILS algorithms on the strengthened formulation, with results categorized by instance set. In the table, the first column presents the instance set with the number of instances in each set (in parentheses). The performance metrics include: the number of instances solved to optimality within the time limit (*#Opt.*), the average running time in seconds (*CPU*), the average percentage optimality gap at termination (*Opt.gap*), and the average number of nodes explored in the B&B tree (*#Node*). Moreover, the '*Overall*' row aggregates results across all instance sets, showing the total number of instances solved to optimality, the grand average running time, the average optimality gap and the average nodes explored. The best performance metrics (excluding *#Node*) for each instance set are highlighted in boldface.

Table 2: Comparison of solution approaches on MTZF (1) and SCFF (2) for instances with $|\Omega| = 10$

	MTZF (1)				strengthened MTZF with (4)-(7) and (9)							
	Gurobi				C&B algorithm				ILS algorithm			
Instance set	#Opt.	CPU	Opt.gap	#Node	#Opt.	CPU	Opt.gap	#Node	#Opt.	CPU	Opt.gap	#Node
C (30)	**30**	23.69	**0.00%**	299.87	**30**	**16.62**	**0.00%**	209.50	**30**	28.78	**0.00%**	710.47
R (30)	19	1488.81	4.64%	14076.93	22	1225.30	1.38%	15158.47	**26**	**814.71**	**0.58%**	69617.71
RC (25)	19	1055.87	4.14%	14074.08	20	**979.48**	1.56%	16195.92	19	992.19	**1.36%**	95164.00
Overall (85)	68	856.13	2.93%	9483.63	72	740.46	0.98%	10521.30	**75**	**611.89**	**0.65%**	55164.06
	SCFF (2)				strengthened SCFF with (4)-(7) and (9)							
	Gurobi				C&B algorithm				ILS algorithm			
Instance set	#Opt.	CPU	Opt.gap	#Node	#Opt.	CPU	Opt.gap	#Node	#Opt.	CPU	Opt.gap	#Node
C (30)	**30**	43.27	**0.00%**	248.90	**30**	**24.34**	**0.00%**	184.47	**30**	30.05	**0.00%**	625.60
R (30)	28	847.58	0.31%	6279.17	28	619.73	0.17%	5203.83	**30**	**584.24**	**0.00%**	61629.50
RC (25)	24	739.18	0.06%	9088.00	**25**	**517.81**	**0.00%**	6506.68	22	743.62	0.58%	58418.16
Overall (85)	82	543.34	0.12%	5205.36	**83**	**387.29**	**0.06%**	3964.99	82	452.64	0.19%	40224.42

From Table 2, we can see that

- strengthening the MTZF yields significant performance gains, enabling more instances to be solved optimally with less computational effort. Similarly, for the SCFF, the strengthened version enables solution approaches, especially the C&B algorithm, to solve more instances to optimality in shorter CPU times;

- on the strengthened MTZF, the ILS algorithm outperforms the C&B algorithm by solving three more instances to optimality with less average CPU time and a lower optimality gap. In contrast, on the strengthened SCFF, the C&B algorithm is superior, solving one more instance optimally with shorter average CPU time, a smaller average optimality gap, and fewer explored nodes than the ILS algorithm;

- and solution approaches based on the SCFF consistently outperform their MTZF counterparts in terms of all performance metrics.

To further examine the scalability of all solution approaches on the SCFF, in the following, we extend our evaluation to instances with expanded scenario sets $|\Omega| \in \{30, 50, 100\}$. The computational results grouped by instance set for each scenario size are presented in Table 3 with dominant performance metrics (excluding *#Node*) emphasized in boldface.

Table 3: Comparison of solution approaches on SCFF (2)

		SCFF (2)				strengthened SCFF with (4)-(7) and (9)							
		Gurobi				C&B algorithm				ILS algorithm			
$\|\Omega\|$	Instance set	#Opt.	CPU	Opt.gap	#Node	#Opt.	CPU	Opt.gap	#Node	#Opt.	CPU	Opt.gap	#Node
30	C (30)	29	340.18	0.12%	468.30	29	217.82	0.06%	261.17	**30**	**119.24**	**0.00%**	910.37
	R (30)	16	2010.62	2.18%	2230.10	19	1727.19	1.66%	2229.17	**23**	**1269.77**	**1.25%**	47374.80
	RC (25)	17	1382.33	1.60%	2052.88	19	1264.84	1.24%	3837.20	**20**	**986.08**	**1.10%**	42969.12
	Overall (85)	62	1244.38	1.30%	1583.76	67	1069.95	0.99%	2109.18	**73**	**791.70**	**0.78%**	30418.10
50	C (30)	28	466.09	0.32%	138.27	29	370.28	0.20%	214.47	**30**	**254.27**	**0.00%**	952.30
	R (30)	14	2289.97	3.14%	953.47	17	2088.34	2.41%	917.50	**21**	**1432.91**	**1.86%**	31684.10
	RC (25)	16	1636.83	2.49%	1178.52	17	1507.33	1.97%	1253.21	**19**	**1135.09**	**1.72%**	31501.31
	Overall (85)	58	1464.30	1.98%	756.75	63	1321.98	1.53%	795.06	**70**	**940.76**	**1.20%**	21379.24
100	C (30)	25	829.59	1.16%	36.57	**27**	**596.62**	0.62%	12.77	**27**	852.28	**0.16%**	615.40
	R (30)	10	2426.71	5.51%	45.83	10	2429.80	4.70%	834.90	**19**	**1820.88**	**2.98%**	9466.03
	RC (25)	12	2231.23	4.63%	803.72	15	1880.74	3.51%	928.77	**18**	**1293.96**	**2.03%**	19964.62
	Overall (85)	47	1829.18	3.77%	295.37	52	1635.72	2.94%	592.14	**64**	**1322.37**	**1.72%**	10015.35

As shown in Table 3, the ILS algorithm consistently outperforms both Gurobi and the C&B algorithm, solving more instances to optimality in a smaller computational time and achieving smaller optimality gaps. Notably, these advantages persist despite the ILS algorithm exploring more nodes in the B&B tree. As the number of scenarios $|\Omega|$ grows from 30 to 100, all solution approaches exhibit a rise in average computational time. Both Gurobi and the C&B algorithm experience a more pronounced performance deterioration, as reflected in the number of optimal solutions found (*#Opt.*), the computational time (*CPU*) and the optimality gap (*Opt.gap*). In contrast, the ILS algorithm demonstrates a superior robustness and a better scalability under increasing scenario counts.

5.2.3. Performance comparison of VRPTW-SC-Insert and VRPTW-SC-Skip

We compare the performance of the optimal solutions obtained for VRPTW-SC-Insert and VRPTW-SC-Skip. To this end, we employ three sets of instances, characterized by a distinct stochastic customer presence probability $p_s = \{0.25, 0.5, 0.75\}$, each instance containing 10 scenarios. The instances with $p_s = 0.25$ and $p_s = 0.75$ are generated as described in Section 5.1.1. Based on the performance of the solution approaches in

Section 5.2.2, we adopt the C&B algorithm applied to the strengthened SCFF to conduct the computational experiments.

Table 4 presents computational results for VRPTW-SC-Insert and VRPTW-SC-Skip, grouped by instance set for each stochastic customer presence probability. We report the number of instances solved to optimality (*#Opt.*), the average percentage optimality gap (*Opt.gap*), the average total cost (optimal or best at termination) (*TCost*), and the average CPU time in seconds (*CPU*). The symbol '–' indicates that results are unavailable, as the problem size renders such large-scale MIPs computationally prohibitive on current workstation hardware due to memory limitations.

Table 4: Performance comparison of VRPTW-SC-Insert and VRPTW-SC-Skip

		VRPTW-SC-Insert				VRPTW-SC-Skip			
p_s	Instance set	#Opt	Opt.gap	TCost	CPU	#Opt	Opt.gap	TCost	CPU
0.25	C (30)	30	0.00%	529.28	6.70	—	—	—	—
	R (30)	30	0.00%	493.81	173.81	23	1.30%	494.77	1350.42
	RC (25)	25	0.00%	598.02	316.21	24	0.16%	599.21	538.20
	Overall (85)	85	0.00%	540.37	165.57	47	0.73%	546.99	944.31
0.5	C (30)	30	0.00%	540.01	24.34	—	—	—	—
	R (30)	28	0.17%	506.12	619.73	28	0.12%	507.43	607.63
	RC (25)	25	0.00%	611.30	517.81	25	0.00%	613.74	213.51
	Overall (85)	83	0.06%	552.48	387.29	53	0.06%	560.59	410.57
0.75	C (30)	30	0.00%	549.39	87.61	—	—	—	—
	R (30)	24	1.02%	517.94	1080.89	25	0.68%	519.66	1270.35
	RC (25)	22	0.51%	627.63	837.74	25	0.00%	628.80	251.13
	Overall (85)	76	0.51%	564.99	668.75	50	0.34%	574.23	760.74

Table 4 reveals that VRPTW-SC-Insert significantly outperforms VRPTW-SC-Skip in terms of tractability, solving 95.7% of all instances to optimality, compared to only 58.8% for VRPTW-SC-Skip, with the latter failing to solve any C-type instances due to the computational burden of incorporating stochastic customers into first-stage routes. Consistent with Proposition 1, VRPTW-SC-Insert achieves lower average total costs across all probability settings. While VRPTW-SC-Insert generally requires less computational effort, an interesting exception occurs on RC instances at higher probabilities ($p_s = 0.5, 0.75$), where VRPTW-SC-Skip solves more instances to optimality in substantially less time, suggesting its structure may occasionally align favorably with random-clustered data characteristics.

5.2.4. Impact of Insertion Path Length

Finally, we examine how flexibility in insertion path length affects solution quality and computational performance for VRPTW-SC-Insert. Specifically, three configurations are considered: (i) unlimited length: paths may contain any number of stochastic customers, (ii) $|p| \leq 2$: each path serves at most two stochastic customers, and (iii) $|p| = 1$: each path serves only one stochastic customer. For each configuration, we perform computational experiments on instances with $|\Omega| = 10$ and $p_s = 0.5$ using the C&B algorithm. Table 5 presents the average computational results, including the total cost (optimal or best at termination) (*TCost*), the number of vehicles used (*#Veh*), and computation time (*CPU*). The results for the unlimited path length case are drawn from Section 5.2.2. For instances where the C&B algorithm did not reach optimality, we substitute the optimal results obtained by the ILS algorithm. All instances under $|p| = 1$ and $|p| \leq 2$ are solved to optimality.

Table 5: Comparison of insertion path length settings

	Unlimited length			$\lvert p\rvert \leq 2$			$\lvert p\rvert = 1$		
instance set	TCost	#Veh	CPU	TCost	#Veh	CPU	TCost	#Veh	CPU
C(30)	540.01	5.33	24.34	540.48	5.27	13.99	543.76	5.30	6.96
R(30)	506.12	5.53	491.00	506.38	5.53	404.43	509.78	5.57	181.24
RC(25)	611.30	5.72	517.81	611.43	5.76	350.38	615.79	5.72	166.31
Overall	552.48	5.53	344.38	552.76	5.52	256.26	556.44	5.53	118.17

Table 5 shows that total cost increases as path length flexibility decreases. The $|p| = 1$ configuration yields the highest cost, while $|p| \leq 2$ achieves near-equivalent quality to the unlimited setting, suggesting that moderate path flexibility enables effective routing consolidation. However, this cost saving comes at a significant computational expense. Compared to $|p| = 1$, the $|p| \leq 2$ setting more than doubles CPU time, while the unlimited configuration takes approximately three times longer. The number of vehicles used remains stable across all configurations, indicating that path flexibility primarily optimizes routing efficiency rather than reducing fleet size.

6. Conclusion

In this study, we investigated an extension of VRPTW that explicitly accounted for stochastic customers, whose uncertainty was modeled using a finite set of scenarios at the planning phase. To tackle this problem, we introduced a classical and a new a priori optimization approaches, formulating the corresponding problems, VRPTW-SC-Skip and VRPTW-SC-Insert, as scenario-based two-stage stochastic programs. We proposed DEFs for both problems: MTZF and SCFF for VRPTW-SC-Insert, and SCFF for VRPTW-SC-Skip. We further proved that the optimal value of VRPTW-SC-Insert is bounded above by that of VRPTW-SC-Skip. To strengthen these formulations, we incorporated lifted SIPCs and SECs, and developed two exact solution methods, C&B and ILS algorithms, to solve the problem.

Computational results for VRPTW-SC-Insert yielded several key insights: 1) incorporating the lifted SIPCs into the formulations resulted in average improvements of 4.28 - 12.28% in the root node LBs. When SECs were further applied, cumulative average improvements increased to 4.81 - 18.42%, with MTZF exhibiting greater benefit than SCFF; 2) strengthening both MTZF and SCFF led to substantial enhancements in computational performance, notably increasing the number of instances solved to optimality while reducing solution times. Among the two, solution methods based on SCFF consistently outperformed their MTZF counterparts across both C&B and ILS algorithms; 3) the ILS algorithm applied to SCFF demonstrated superior robustness and scalability as the number of scenarios increased, outperforming both Gurobi and the C&B algorithm; 4) limited path flexibility captured the majority of benefits afforded by the recourse strategy; and 5) VRPTW-SC-Insert is more tractable than VRPTW-SC-Skip, while consistently yielding lower average total costs.

This study opens up several avenues for future research. An extension would be to incorporate a dedicated backup vehicle stationed at depot to serve stochastic customers, a common practice in healthcare operations. This can be modeled by including the arc $(0, 0)$ with $c_{00} = t_{00} = 0$ in the arc set, thereby allowing the assignment of realized stochastic customers to the dedicated vehicle as an additional recourse action. From the methodological point of view, it may be interesting to not include the variables for all the possible paths in advance, rather generate them when they are needed, to be able to solve instances with larger number of stochastic customers.

Acknowledgments

We would like to thank Prof. Erik Demeulemeester and Prof. Roel Leus for their valuable comments and suggestions on a preliminary version of this work.

References

All I.T. A Guide to Understanding Onsite IT Support and Field Service Solutions. Accessed October 15, 2025. https://www.allitsupported.com/a-guide-to-understanding-onsite-it-support-and-field-service-solutions/.

Angulo, G., Ahmed, S., & Dey, S. S. (2016). Improving the integer L-shaped method. INFORMS Journal on Computing, 28(3), 483-499.

Ascheuer, N., Fischetti, M., & Grötschel, M. (2000). A polyhedral study of the asymmetric travelling salesman problem with time windows. Networks, 36(2), 69–79.

Ascheuer, N., Fischetti, M., & Grötschel, M. (2001). Solving the asymmetric travelling salesman problem with time windows by branch-and-cut. Mathematical Programming, 90(3), 475-506.

Avella, P., Boccia, M., & Vasilyev, I. (2013). Lifted and local reachability cuts for the vehicle routing problem with time windows. Computers & Operations Research, 40(8), 2004-2010.

Baldacci, R., Mingozzi, A., & Roberti, R. (2011). New route relaxation and pricing strategies for the vehicle routing problem. Operations Research, 59(5), 1269-1283.

Bard, J. F., Kontoravdis, G., & Yu, G. (2002). A branch-and-cut procedure for the vehicle routing problem with time windows. Transportation Science, 36(2), 250-269.

Bektaş, T., Repoussis, P. P., & Tarantilis, C. D. (2014). Chapter 11: Dynamic vehicle routing problems. In Vehicle Routing: Problems, Methods, and Applications, Second Edition (pp. 299-347). Society for Industrial and Applied Mathematics.

Benders, J. (1962). Partitioning procedures for solving mixed-variables programming problems. Numerische Mathematik, 4(1), 238–252.

Boland, N., Dethridge, J., & Dumitrescu, I. (2006). Accelerated label setting algorithms for the elementary resource constrained shortest path problem. Operations Research Letters, 34(1), 58-68.

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(3), 387-404.

Desrochers, M., Desrosiers, J., & Solomon, M. (1992). A new optimization algorithm for the vehicle routing problem with time windows. Operations Research, 40(2), 342-354.

Gavish, B., & Graves. S. (1979). The traveling salesman problem and related problems. Working Paper, Graduate School of Management, University of Rochester, New York.

Gendreau, M., Laporte, G., & Séguin, R. (1995). An exact algorithm for the vehicle routing problem with stochastic demands and customers. Transportation Science, 29(2), 143-155.

Gendreau, M., Jabali, O., & Rei, W. (2014). Chapter 8: Stochastic vehicle routing problems. In Vehicle Routing: Problems, Methods, and Applications, Second Edition (pp. 213-239). Society for Industrial and Applied Mathematics.

ISG. Manufacturing Field Service Buyers Guide 2025 Executive Summary. Accessed October 15, 2025. http://research.isg-one.com/buyers-guide/customer_experience/manufacturing_field_service/2025.

Irnich, S., Desaulniers, G., Desrosiers, J., & Hadjar, A. (2010). Path-reduced costs for eliminating arcs in routing and scheduling. INFORMS Journal on Computing, 22(2), 297-313.

Jaillet, P. (1985). Probabilistic traveling salesman problems (Doctoral dissertation, Massachusetts Institute of Technology).

Jaillet, P. (1988). A priori solution of a traveling salesman problem in which a random subset of the customers are visited. Operations Research, 36(6), 929-936.

Jepsen, M., Petersen, B., Spoorendonk, S., & Pisinger, D. (2008). Subset-row inequalities applied to the vehicle-routing problem with time windows. Operations Research, 56(2), 497-511.

Kallehauge, B., Boland, N., & Madsen, O. B. (2007). Path inequalities for the vehicle routing problem with time windows. Networks, 49(4), 273-293.

Kallehauge, B., Larsen, J., & Madsen, O. B. (2006). Lagrangian duality applied to the vehicle routing problem with time windows. Computers & Operations Research, 33(5), 1464-1487.

Kohl, N., Desrosiers, J., Madsen, O. B., Solomon, M. M., & Soumis, F. (1999). 2-path cuts for the vehicle routing problem with time windows. Transportation Science, 33(1), 101-116.

Kohl, N., & Madsen, O. B. (1997). An optimization algorithm for the vehicle routing problem with time windows based on Lagrangian relaxation. Operations Research, 45(3), 395-406.

Lagos, F., Klapp, M. A., & Toriello, A. (2023). Branch-and-price for routing with probabilistic customers. Computers & Industrial Engineering, 183, 109429.

Laporte, G., & Louveaux, F. V. (1993). The integer L-shaped method for stochastic integer programs with complete recourse. Operations Research Letters, 13(3), 133-142.

Laporte, G., Louveaux, F. V., & Mercure, H. (1994). A priori optimization of the probabilistic traveling salesman problem. Operations Research, 42(3), 543-549.

Letchford, A. N., & Salazar-Gonzalez, J. J. (2006). Projection results for vehicle routing. Mathematical Programming, 105, 251-274.

Lysgaard, J. (2006). Reachability cuts for the vehicle routing problem with time windows. European Journal of Operational Research, 175(1), 210-223.

Miller, C. E., Tucker, A. W., & Zemlin, R. A. (1960). Integer programming formulation of traveling salesman problems. Journal of the ACM, 7(4), 326-329.

Gotadki, R. (2021). Home Healthcare Market Research Report: Size, Share, Trend Analysis By Product, By Service, By Indication, And By Region – Market Forecast Till 2035. Accessed October 15, 2025. https://www.marketresearchfuture.com/reports/home-healthcare-market-2030.

Oracle. What Is Field Service?. Accessed October 15, 2025. https://www.oracle.com/cx/service/field-service-management/what-is-field-service/.

Özarik, S. S., Lurkin, V., Veelenturf, L. P., Van Woensel, T., & Laporte, G. (2023). An adaptive large neighborhood search heuristic for last-mile deliveries under stochastic customer availability and multiple visits. Transportation Research Part B: Methodological, 170, 194-220.

Ozbaygin, G., Yaman, H., & Karasan, O. E. (2016). Time constrained maximal covering salesman problem with weighted demands and partial coverage. Computers & Operations Research, 76, 226-237.

Pecin, D., Contardo, C., Desaulniers, G., & Uchoa, E. (2017). New enhancements for the exact solution of the vehicle routing problem with time windows. INFORMS Journal on Computing, 29(3), 489-502.

Righini, G., & Salani, M. (2006). Symmetry helps: Bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. Discrete Optimization, 3(3), 255-273.

Righini, G., & Salani, M. (2008). New dynamic programming algorithms for the resource constrained elementary shortest path problem. Networks, 51(3), 155-170.

Rousseau, L. M., Gendreau, M., & Feillet, D. (2007). Interior point stabilization for column generation. Operations Research Letters, 35(5), 660-668.

Rousseau, L. M., Gendreau, M., Pesant, G., & Focacci, F. (2004). Solving VRPTWs with constraint programming based column generation. Annals of Operations Research, 130, 199-216.

Sherali, H. D., & Lunday, B. J. (2013). On generating maximal nondominated Benders cuts. Annals of Operations Research, 210, 57-72.

SNS Insider. (2023). Field Service Management (FSM) Market Size, Share & Segmentation By Component, By Deployment, By Enterprise, By End-use, By Region, and Global Forecast 2024-2032. Accessed October 15, 2025. https://www.snsinsider.com/reports/field-service-management-market-2888.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations Research, 35(2), 254-265.

Spoorendonk, S., & Desaulniers, G. (2010). Clique inequalities applied to the vehicle routing problem with time windows. INFOR: Information Systems and Operational Research, 48(1), 53-67.

Waters, C. D. J. (1989). Vehicle-scheduling problems with uncertainty and omitted customers. Journal of the Operational Research Society, 40(12), 1099-1108.

Wissink, P. L. (2023). The traveling salesman problem with stochastic and correlated customers. Transportation Science, 57(5), 1321-1339.