

# Column Generation-Based Solution Approach for a Stochastic Vehicle Routing Problem with Flexible Deliveries

İhsan Sadati<sup>a,b,\*</sup> and Duygu Taş<sup>a,b</sup>

<sup>a</sup> Sabanci University, Faculty of Engineering and Natural Sciences, Orta Mahalle, 34956 Tuzla, Istanbul, Türkiye

<sup>b</sup> Smart Mobility and Logistics Lab, Sabanci University, Orta Mahalle, 34956 Tuzla, Istanbul, Türkiye

**Abstract:** This paper introduces a stochastic vehicle routing problem with flexible delivery locations and soft time window constraints. In this problem, customers are served at one of their preferred delivery locations, and travel times are uncertain. We propose a mathematical model in which the objective is to minimize a weighted combination of operational costs and service-related costs. To obtain a solution to the problem, we develop a column generation-based solution approach that obtains efficient routes via the pricing subproblem which corresponds to an elementary shortest path problem with resource constraints. This subproblem is effectively solved by implementing the ng-route relaxation technique and a parallelized label correcting algorithm. The proposed method initializes a feasible solution for a set of routes, which is then used to construct the restricted linear programming master problem of column generation. The linear relaxation provides a lower bound when column generation terminates, and an upper bound is obtained by solving an integer programming model over all efficient routes obtained by the column generation. The computational experiments conducted on well-known problem instances demonstrate that the approach provides reasonable optimality gaps and computational times. We further adapt the solution approach for an extension that adds shared delivery locations resulting in cost savings. Overall, the results show that the proposed solution approach effectively balances cost efficiency and service reliability for flexible delivery routing in stochastic environment.

**Keywords:** Vehicle routing; Flexible deliveries; Stochastic travel times; Column generation.

## 1. Introduction

The growth of e-commerce has significantly complicated last-mile delivery operations, leading logistics providers to adopt more flexible and customer-focused approaches. For example, “DHL parcel’s multiple delivery options program” allows customers to redirect or reschedule deliveries through several channels (DHL, 2026), while “UPS access point services” enable the collection of parcels from nearby partner stores or lockers at a convenient time (UPS, 2026). These developments reflect a growing shift in urban logistics toward greater delivery flexibility and customer input. Consequently, they have motivated research into optimization models for routing problems in which customers may be served at one of several potential delivery locations. From a modeling perspective, this emerging delivery paradigm can be represented through the Vehicle Routing Problem with Flexible Deliveries (VRPFD),

---

\* Corresponding author: msadati@sabanciuniv.edu

an extension of the classical Vehicle Routing Problem with Time Windows (VRPTW). The VRPFD allows each customer to be served at one of several potential delivery locations, each associated with a distinct time window. This flexibility captures new delivery practices where customers may choose among alternative locations, such as home, workplace, or designated collection points, depending on daily circumstances. This line of research has also been explored in the Electric VRP (EVRP) literature. In particular, Sadati et al. (2022) study an EVRPFD with depot-only recharging, while Duman et al. (2024) extend this setting to allow en-route recharging and developed a branch-and-price solution approach.

In real-life urban delivery operations, limited deviations from delivery time bounds are usually accepted to improve routing efficiency while maintaining service reliability (see, e.g., Zhang et al., 2013; Taş et al., 2014a.; Taş et al., 2014b). Early or late service is therefore permitted, typically with an associated penalty cost. Similarly, driver overtime is often allowed in terms of extended working hours, incurring additional costs to ensure route completion when schedules cannot be strictly met. Earliness arises when a vehicle begins service before the earliest allowable start time of the customer, rather than waiting until the preferred time window opens. In such cases, the service starts prematurely, violating the intended schedule. This deviation does not stem from driver waiting time; instead, the penalty reflects the operational and customer-service disadvantages of initiating service earlier than planned. Accordingly, an earliness cost is imposed to discourage routes that require service to start too early. Lateness (or delay) refers to serving a customer after the time window ends, which decreases customer satisfaction and incurs a lateness penalty proportional to the expected delay duration. Overtime represents the excess of a driver's working time beyond the regular shift length, introducing a cost component that accounts for operational-related expenses. Such operational flexibility is particularly important given that travel times are rarely deterministic, since random events (such as accidents) may disrupt planned schedules. Modeling travel time as a random variable allows planners to capture expected earliness, lateness (delay), and overtime effects that arise from stochastic variations in the time required to traverse connections.

In this study, we introduce a Stochastic Vehicle Routing Problem with Flexible Deliveries (SVRPFD), which extends the flexible-delivery paradigm by explicitly incorporating stochastic travel times. Our formulation extends the VRPFD by integrating stochastic travel time distributions and soft time-window constraints into a unified optimization framework. Specifically, each customer can be served at one of several potential delivery locations (e.g., home, workplace, and so on), each with its own soft time window. The model selects the most cost-effective combination of delivery locations and routes by accounting for both operational costs and service costs. The latter include expected earliness, lateness, and driver overtime under travel times assumed to follow a known probability distribution, specifically the Gamma distribution (see, e.g., Taş et al., 2013 and Taş et al., 2014). Notably, our formulation allows overlapping time windows across different delivery locations of the same customer, a feature that has received limited attention in prior flexible-delivery studies, which often consider alternative locations with distinct or non-overlapping availability windows (Dragomir et al., 2022). We

then propose a solution approach based on column generation to solve the problem effectively. The method employs techniques such as ng-route relaxation and parallelized label-correcting algorithms to accelerate convergence and enhance computational efficiency. We conduct computational experiments on a range of instances to evaluate the model and algorithm. Moreover, we consider further practical extensions: shared delivery scenario where customers located near each other can be served jointly at a common location. The results provide insights into the cost benefits of flexible and shared deliveries under uncertainty and highlight the trade-offs and parameter sensitivities involved.

The remainder of this paper is organized as follows. Section 2 reviews relevant literature on stochastic VRPs, and flexible and shared delivery routing. Section 3 presents the mathematical formulation of the base problem (SVRPFD) and its extension with Shared Delivery Locations (SVRPFD-SDL). Section 4 details the column generation algorithm developed to solve the problem. Section 5 reports computational results for both the SVRPFD and the SVRPFD-SDL, with discussion. Section 6 offers concluding remarks and future research directions.

## **2. Literature Review**

With the growth of e-commerce and its impact on last-mile logistics, there have been major advancements in the VRP literature relevant to flexible and alternative delivery options. This study draws from two overarching areas of relevant literature: deterministic models with flexible and/or roaming deliveries, and stochastic models that specifically consider travel time uncertainty.

### **2.1. Deterministic Models with Flexible or Roaming Deliveries**

Reyes et al. (2017) introduced the VRPRDL and provided a mathematical model that also covers a home-plus-roaming variant. The authors designed route-construction and improvement heuristics with two problem-specific ideas: (i) optimizing each customer's selected delivery location along a route, and (ii) effectively changing neighbors' locations when removing a customer from a route. Using two benchmark sets that do not enforce triangle inequality, they reported distance reductions up to 65% when customers can declare multiple delivery locations. It is worth noting that we also incorporate home-plus-roaming variant into our computational study to examine how the availability of a home option affects the solution quality and computational time. To exactly solve the VRPRDL, Ozbaygin et al. (2017) proposed a branch-and-price method. They adapted the problem instances developed by Reyes et al. (2017) to satisfy the triangle inequality and introduced 20 new instances to the literature. Their approach obtained optimal solutions for 60-customer instances in minutes, and feasible solutions for 120-customer instances with a gap at a six-hour time limit.

A dynamic VRPRDL, where daily schedule changes may alter feasible time windows and necessitate re-optimization, was later handled with a branch-and-price-based updating scheme by Ozbaygin and Savelsbergh (2019). For a multi-depot variant with strict time windows, Jolfai et al. (2024) combined evolutionary local search with Large Neighborhood Search (LNS) and showed that flexible delivery policies with multi-depot planning can reduce the total traveled distance by up to 19% on large cases (e.g., 288 customers, 6 depots, 30 vehicles). Dumez et al. (2021) introduced flexible deliveries that

generalize VRPRDL by allowing many delivery alternatives (home, trunk, and parcel lockers). The authors proposed an LNS algorithm whose route-recombination phase periodically solves cluster partitioning subproblems, producing feasible solutions to the instances with up to 200 customers and 400 options. Tilk et al. (2021) provided an exact branch-and-cut method where they solved the pricing subproblem using a bi-directional ng-route algorithm. The authors created new instances with up to 100 options, obtained proven optima for the instances with up to 50 customers and 100 options, and settled long-open optimal solutions for 17 VRPRDL instances previously unsolved by Ozbaygin et al. (2017). Pourmohammadreza and Jokar (2024) addressed VRP with delivery options through a two-stage approach: multi-criteria evaluation of candidate delivery points, followed by a mixed-integer model for routing and timing. With real data, the authors reported more than 25% cost reductions when delivery options are selected strategically. Janinhoff et al. (2024) studied multi-trip VRPs with delivery options using large real networks (892 urban and 1101 rural customers). An adaptive large neighborhood search provided cost decreases of 3.6% in urban and 4.4% in rural cases, where multi-trip operation also reduced fleet size and deadheading. Mancini and Gansterer (2021) considered an either home or shared point model (e.g., parcel lockers). Since home is preferred by customers yet costly for carriers, the model assumes compensating payments when assigning a shared location. The proposed matheuristic approach effectively solved instances with up to 75 customers and 5 shared points, demonstrating good solution quality within reasonable computational times. Frey et al. (2023) examined a related problem where each customer is served at exactly one location, and each location has limited capacity. Unlike flexible deliveries or shared delivery locations, this variant excludes home delivery and customer preferences.

Flexible delivery options have recently been integrated into electric vehicle routing. Sadati et al. (2022) introduced an Electric VRP (EVRP) with flexible deliveries where vehicles recharge only at the depot and developed a hybrid variable neighborhood tabu search approach to generate efficient solutions. Duman et al. (2024) extended to en-route recharging and proposed a branch-and-price algorithm tested on new instances with up to 120 customers. The authors reported proven optima for 53 out of 60 instances. For the home-included variant, the authors provided valid upper bounds for 33 out of 60 instances via an integer programming at the root node when the time limit was reached.

## **2.2. Stochastic Extensions with Uncertain Travel Times**

Routing models that explicitly incorporate travel-time uncertainty have received increasing attention in the operations research literature. Early stochastic vehicle routing studies, such as Laporte et al. (1992), investigated the effect of random travel durations on route planning, laying the foundation for subsequent research on uncertainty in last-mile logistics. Building upon this foundation, Taş et al. (2013) developed one of the first stochastic VRP models with soft time windows and random travel times. The authors proposed a tabu search heuristic enhanced by a dedicated initialization and post-optimization procedure to generate high-quality routes efficiently under travel-time uncertainty. Later, Taş et al. (2014a) advanced this framework by introducing a branch-and-price algorithm, where column generation is solved at each node of the tree. Their study explicitly incorporated stochastic travel times

in route evaluation and quantified expected penalties, producing reliable solutions even under significant travel-time variability. More recent contributions further enhance realism by accounting for time-dependent (Taş et al, 2014b) or correlated travel times (Bomboi et al., 2021), network-wide reliability (Rajabi-Bahaabadi et al., 2021) and data-driven distribution estimation for deadlines and routing (Wang et al., 2024).

Although the body of research continues to grow, studies addressing both multiple delivery-location options per customer and stochastic travel-time modeling remain relatively scarce. Lombard et al. (2018) studied a stochastic VRPRDL by replacing deterministic travel times with random variables of known distributions and minimizing expected route duration. A GRASP integrated with Monte Carlo simulation solved instances with up to 30 customers. It is observed that even small-sized instances can be computationally difficult when stochastic traveling times are considered (e.g., a six-customer case could be solved in two hours). Sampaio et al. (2019) modeled each arc time as the sum of a deterministic component and a random disruption. The authors solved this problem via a two-stage stochastic program: deterministic routing first (early arrivals are not permitted), then recourse that accounts for customers missing their windows (served by third-party logistics with a penalty) and for depot returns after horizon closure (overtime capped at two hours). A sample average approximation was developed and the method obtained feasible solutions for instances with up to 30 customers with at most five locations per customer. He et al. (2020) addressed a VRPRDL with stochastic travel times, formulated as a two-stage stochastic programming model with recourse. Their objective was to minimize the total travel time for serving customers under travel-time uncertainty. To solve the model, the authors proposed a metaheuristic combining a random selection-greedy insertion phase and a hybrid iterated greedy algorithm with route re-optimization and simulated annealing. Their developed heuristic obtained optimal solutions for four out of 10 instances with at most 20 customers, whereas the construction heuristic of Reyes et al. (2017) could not obtain any optimal solution on those cases. In addition, they reported optimal solutions for 11 out of 20 small- and medium-sized instances with at most 30 customers, compared with 5 out of 20 optimal solutions obtained by the improvement heuristic of Reyes et al. (2017).

### **2.3. Research Gap and Contributions**

While several deterministic versions of the VRPFD have been studied extensively, work on its stochastic variants has been limited in both modeling and computational approach. Existing stochastic VRPs that incorporate delivery options or roaming flexibility are often simplified for tractability, rather than modeled using full probability distributions. Additionally, many stochastic studies provide a single set of results on relatively small-sized instances. This suggests that there is a need for more integrated models that jointly characterize travel time uncertainty and service flexibility, time-window deviations, and within a scalable optimization method. This study addresses the gap in the literature by introducing a stochastic VRP with flexible delivery service and soft time windows, where travel times are assumed to be uncertain and follow a known probability distribution. In addition, we consider two extensions of the proposed problem, namely the home roaming delivery locations and the shared delivery location

case. To solve these problems effectively, we adopt a column-generation framework that integrates the ng-route relaxation within a pricing subproblem corresponding to an elementary shortest path problem with resource constraints. To enhance the computational scalability of the pricing subproblem, we further employ a parallel implementation of label-correcting and dominance rules, enabling faster convergence to efficient routes within the stochastic problem setting.

### 3. Mathematical Formulation

This section presents the mathematical formulations developed for the proposed SVRPFD (Section 3.1), as well as its extension that considers shared delivery locations (Section 3.2). The SVRPFD integrates both uncertainty in travel times and flexibility in delivery locations, enabling realistic modeling of customer-level flexibility and travel time uncertainty. In the extension with Shared Delivery Locations (SDLs), multiple customers can be served at common delivery points such as parcel lockers or neighborhood pickup centers. This extension captures practical delivery network consolidation strategies while preserving service flexibility.

#### 3.1. Stochastic Vehicle Routing Problem with Flexible Deliveries

We consider the SVRPFD defined on a directed graph  $G = (V, A)$ , where  $V = \{0\} \cup N$  represents the set of all nodes and  $A$  denotes the set of directed arcs between them. Node 0 corresponds to the central depot, and the remaining nodes represent delivery locations. Each customer  $c \in C$  has a set of alternative delivery locations  $N_c \subseteq N$ , which may include the customer's home, workplace, or a nearby location. A delivery to a customer can be performed at exactly one of these potential locations, and the decision of which location to select is part of the optimization process. Each customer  $c$  has a demand  $q_c$ , representing the required quantity to be delivered. Each delivery location  $i \in N$  is associated with a delivery time window  $[e_i, l_i]$  and a service time  $b_i$ . These time windows are soft, meaning deliveries can occur outside the interval, but this results in additional earliness or lateness penalties. The depot has its own time window  $[e_0, l_0]$ , defining the earliest departure and latest return times for all vehicles. Each driver has a working time limit  $w$ , representing the shift duration. The travel distance along each arc  $(i, j) \in A$  is denoted by  $d_{ij}$ . Each used vehicle  $k \in K$  incurs a fixed cost  $c_f$ . The parameters  $c_t$  and  $c_o$  denote the unit travel cost and the unit overtime cost, respectively. Additionally, penalty coefficients  $c_e$  and  $c_l$  are assigned per unit of the expected earliness and lateness, respectively. All vehicles are assumed to be identical with a capacity  $Q$ .

**Table 1.** Decision Variables used in the Proposed Formulation

Decision Variables	
$x_{ijk}$	Binary variable taking the value of 1 if vehicle $k \in K$ travels from node $i$ to $j$ , $(i, j) \in A$
$y_{ck}$	Remaining load of vehicle $k \in K$ after serving customer $c \in C$
$\mathbf{x}$	Vector of vehicle assignments and location sequences in these vehicle routes, where $\mathbf{x} = \{x_{ijk} \mid (i, j) \in A, k \in K\}$
$E_{jk}(\mathbf{x})$	Expected earliness at node $j \in N$ when it is visited by vehicle $k \in K$
$L_{jk}(\mathbf{x})$	Expected lateness at node $j \in N$ when it is visited by vehicle $k \in K$
$O_k(\mathbf{x})$	Expected overtime of vehicle $k \in K$

Table 1 presents the decision variables used in the problem formulation. Following Taş et al. (2013, 2014a), we assume that travel times between nodes are Gamma distributed, leading to Gamma distributed arrival times. Consequently, the expected components (earliness, lateness, and overtime) are computed exactly. The mathematical model proposed for the SVRPFD can then be formulated as follows:

$$\min \mu \left( c_t \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} d_{ij} x_{ijk} + c_f \sum_{j \in V \setminus \{0\}} \sum_{k \in K} x_{0jk} + c_o \sum_{k \in K} O_k(\mathbf{x}) \right) + (1 - \mu) \left( c_e \sum_{j \in N} \sum_{k \in K} E_{jk}(\mathbf{x}) + c_l \sum_{j \in N} \sum_{k \in K} L_{jk}(\mathbf{x}) \right) \quad (1)$$

$$\sum_{i \in N_c} \sum_{j \in V \setminus N_c} \sum_{k \in K} x_{ijk} = 1, \quad c \in C, \quad (2)$$

$$\sum_{i \in V} x_{ihk} = \sum_{j \in V} x_{hjk}, \quad h \in N, k \in K, \quad (3)$$

$$\sum_{j \in N} x_{0jk} = 1, \quad k \in K, \quad (4)$$

$$\sum_{i \in N} x_{i0k} = 1, \quad k \in K, \quad (5)$$

$$0 \leq y_{ck} \leq Q - q_c, \quad c \in C, k \in K, \quad (6)$$

$$y_{ck} + Q \left( 1 - \sum_{i \in N_c} \sum_{j \in N_{c'}} x_{ijk} \right) \geq q_{c'} + y_{c'k}, \quad c \in C \cup \{0\}, c' \in C \setminus \{c\}, k \in K, \quad (7)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijk} \leq |B| - 1, \quad B \subseteq N, k \in K, \quad (8)$$

$$x_{ijk} \in \{0,1\}, \quad i \in V, j \in V, k \in K. \quad (9)$$

The objective (1) is to minimize the total weighted cost, consisting of two major components: operational costs and service-level penalties. The operational component includes total travel costs, fixed vehicle usage costs, and expected overtime costs, while the service-level component accounts for the costs incurred due to the expected earliness and lateness of deliveries. The weight  $\mu \in [0,1]$  allows decision-makers to balance cost efficiency against customer service quality. Constraints (2) ensure that each customer is served exactly once at one of its available locations. Constraints (3) enforce flow conservation for each vehicle, ensuring that every visited node has exactly one incoming and one outgoing arc traversed. Constraints (4) and (5) ensure that each vehicle route starts and ends at the depot. Constraints (6) maintain feasible loading by ensuring that vehicle capacity is not exceeded. Constraints (7) link load variables between consecutive nodes on a route, preventing infeasible deliveries. Constraints (8) eliminate subtours, guaranteeing route continuity. Finally, constraints (9) specify that partial services are not permitted. This base model provides the foundation for further extensions, including shared delivery locations discussed next.

### 3.2. SVRPFD with Shared Delivery Locations

In practice, certain delivery nodes, such as parcel lockers or shared access points, can serve multiple customers simultaneously. This concept, referred to as shared delivery locations (SDLs), has been

studied as a means of balancing service convenience with cost efficiency in last-mile logistics. Zhang and Lee (2016) explored shared reception boxes to enhance flexibility by mitigating time-window constraints of traditional home delivery. Similarly, Mancini and Gansterer (2021) analyzed the integration of private and shared delivery options, showing that combining attended home delivery with SDLs can significantly reduce transportation costs without compromising perceived service quality. Building on these insights, our model incorporates SDLs as an extension of the base SVRPFDP, enabling vehicles to serve multiple customers at common delivery nodes within the same route. We refer to this problem as SVRPFDP-SDL. In this extension, each SDL can accommodate deliveries for up to  $m$  customers. When a shared location is visited, all customers assigned to it are considered served, and their individual delivery locations are excluded from further routing. Table 2 provides additional mathematical notations for the extended SDL problem.

**Table 2.** Additional mathematical notations for the SDL Model

<i>Sets</i>	
$S \subseteq N$	Set of shared delivery locations (SDLs)
$P(s) \subseteq C$	Set of customers assigned to SDL $s \in S$
$S_c$	Set of SDLs that can serve customers $c \in P(s)$ where $s \in S$
$L(c) = N_c \cup S_c$	Set of delivery locations for customer $c \in C$
<i>Parameters</i>	
$q_i$	Demand of location $i$ : $q_i = \begin{cases} q_c & \text{if } i \in N_c \\ \sum_{c \in A(i)} q_c & \text{if } i \in S_c \\ 0 & \text{if } i = 0 \end{cases}$
<i>Decision Variables</i>	
$u_{ik}$	Binary variable taking the value of 1 if vehicle $k \in K$ visit node $i \in N$
$h_s$	Binary variable taking the value of 1 if SDL $s \in S$ is visited by one vehicle

The extended mathematical model can be provided as follows:

$$\min \mu \left( c_t \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} d_{ij} x_{ijk} + c_f \sum_{j \in V \setminus \{0\}} \sum_{k \in K} x_{0jk} + c_o \sum_{k \in K} O_k(\mathbf{x}) \right) + (1 - \mu) \left( c_e \sum_{j \in N} \sum_{k \in K} E_{jk}(\mathbf{x}) + c_l \sum_{j \in N} \sum_{k \in K} L_{jk}(\mathbf{x}) \right) \quad (10)$$

(3)-(9)

$$\sum_{i \in N_c} \sum_{j \in V \setminus N_c} \sum_{k \in K} x_{ijk} + \sum_{s \in S_c} h_s = 1, \quad c \in C, \quad (11)$$

$$u_{ik} = \sum_{j \in V} x_{ijk} = \sum_{j \in V} x_{jik}, \quad i \in N, k \in K, \quad (12)$$

$$\sum_{k \in K} u_{sk} = h_s, \quad s \in S, \quad (13)$$

$$u_{ik} \in \{0,1\}, \quad i \in N, k \in K, \quad (14)$$

$$h_s \in \{0,1\}, \quad o \in O. \quad (15)$$

Constraints (11) indicate that each customer is served either at a personal location or through an SDL. Constraints (12) activate the shared location only if at least one assigned customer is served through it, linking customer assignments to route decisions. Constraints (13) prevent multiple vehicles from visiting the same SDL by ensuring that each SDL can be visited by at most one vehicle. Constraints

(14) and (15) denote the domain of the decision variables. Note that the routing, capacity, and time-related constraints remain consistent with the base SVRPFD formulation. This extended formulation captures realistic operational behaviors where customers may prefer to collect deliveries from nearby shared facilities. Incorporating SDLs may allow the model to reduce total travel costs and route lengths in dense urban areas, while maintaining flexibility and service quality. The resulting model integrates both individual and shared delivery mechanisms, enabling the analysis of their trade-offs in terms of cost, service, and network utilization.

#### 4. Solution Methodology

The proposed solution methodology is a column generation-based heuristic (CGH), where we first concentrate on generating efficient columns via column generation and then solve an integer programming model over all these columns. We next present the master problem and the pricing subproblem proposed for the formulation given in Section 3.1.

##### 4.1. Master Problem

We require that each customer  $c$  is covered exactly once (see Constraints (2) in the SVRPFD formulation). Let  $R$  be the set of all feasible routes. A binary variable  $z_r$  is associated with each route  $r \in R$ , indicating whether route  $r$  is selected in the solution (1 if selected, 0 otherwise). Each route  $r$  has a known cost  $f_r$  computed with respect to the objective (1). The master problem is then formulated as follows:

$$\min f_r z_r \tag{16}$$

$$\sum_{r \in R} \sum_{i \in N_c} a_{ir} z_r = 1, \quad c \in C, \tag{17}$$

$$z_r \in \{0,1\}, \quad r \in R. \tag{18}$$

Here  $a_{cr}$  is a parameter equal to 1 if route  $r$  serves customer  $c$ , and 0 otherwise. Constraints (17) ensure each customer is covered by exactly one of the chosen routes.

##### 4.2. Pricing Subproblem

The pricing subproblem for each vehicle  $k$ , which corresponds to constraints (3)-(9) in the SVRPFD formulation, is an Elementary Shortest Path Problem with Resource Constraints (ESPPRC). The pricing subproblem can be formulated as follows:

$$\min \bar{f}_r \tag{19}$$

Subject to (3)-(9)

In the above formulation,  $r$  represents the route of vehicle  $k$ , and  $\bar{f}_r$  is the reduced cost of route  $r$ . For each potential route  $r$ , the reduced cost is calculated as follows:

$$\bar{f}_r = f_r - \sum_{c \in C} \sum_{i \in N_c} a_{ir} \pi_c, \tag{20}$$

where  $\pi_c, c \in C$  is the dual variable associated with the coverage constraints (17). In our problem setting, only vehicle capacity is considered as a resource, since violations of time windows are permitted with associated penalty costs. In the CGH framework, constraints (17) are relaxed, yielding a linear

programming formulation of the master problem. In addition, the master problem initially includes only the vehicle routes of a feasible starting solution and is therefore referred to as the Restricted Linear Programming Master Problem (RLPMP). At each iteration of the column generation algorithm, optimal dual values are obtained by solving the RLPMP and the pricing subproblem is then solved using these dual values to identify new efficient routes. Next, the RLPMP is reoptimized with the addition of these routes. The algorithm terminates when the pricing subproblem fails to generate new efficient routes, indicating that an optimal solution to the linear relaxation of (16)-(18) has been reached.

To obtain a set of feasible routes to initialize the RLPMP, we employ a greedy nearest neighbor heuristic adapted to the flexible-delivery structure. The procedure operates with two dynamic lists: (i) a list of customers who have not yet been served, and (ii) a list of delivery locations that have not been visited. A route is constructed step by step, beginning at the depot. At each iteration, the algorithm selects the closest location to the most recently visited node, provided that adding this location does not violate the vehicle capacity constraint. If the closest location would exceed capacity, the algorithm considers the next-closest alternative, and so forth. When no further location can be feasibly added, the vehicle returns to the depot, and the route is terminated. A key aspect of the heuristic is the handling of customers with multiple delivery locations. When a location is added to the current route, the corresponding customer is marked as served (removed from the unserved-customer list) and all of that customer's alternative delivery locations are removed from the unvisited-location list. This ensures that each customer appears exactly once in the initial solution. New routes are initiated until all customers are assigned to exactly one delivery location and incorporated into a route. Although simple, this greedy nearest neighbor construction provides a fast way to obtain a fully feasible set of starting routes, covering every customer exactly once and respecting vehicle capacity.

### **4.3. Solving the Pricing Subproblem**

We solve the pricing subproblem using a label-correcting algorithm, based on the well-known approach proposed by Feillet et al. (2004) for the ESPPRC. We generate routes by extending partial paths from customer location to customer location, tracking the total cost and resource consumption. Each partial route ending at some customer location  $v$  with a certain residual vehicle capacity is represented by a label. When extending a partial route, we only consider feasible moves, i.e., the vehicle capacity cannot be exceeded. If adding a particular next customer location would make the route infeasible, that extension is not allowed. If it is feasible, we compute the new total cost and resource consumptions for the extended route.

To enforce the condition that each customer is served at most once within a route, a location-based reachability mechanism inspired by Feillet et al. (2004) is employed. When a location associated with a given customer is visited in a partial route, that customer is considered served, and all remaining alternative locations of the same customer are marked as unreachable for further route extensions. A location may become unreachable because (i) it has already been visited, or (ii) extending the route to that location would violate feasibility conditions (e.g., capacity), (ii) or the customer has already been served at another location. This approach ensures elementarity at the customer level, while allowing the

pricing algorithm to explicitly handle customers with multiple potential delivery locations. In the next subsection, this strict customer-level elementarity mechanism is relaxed using the ng-route technique.

#### 4.3.1. Dominance Rules

The pricing subproblem is formulated as an ESPPRC and is solved using a label-correcting algorithm. Without an effective dominance mechanism, the number of labels can grow exponentially. Dominance rules, therefore, play a central role that eliminating partial routes that can never lead to an optimal or improving full route, ensuring that only promising labels are kept throughout the extension process. In the context of the ESPPRC, each partial route  $r$  reaching a node  $i$  is represented by a label vector together with its reduced cost  $\bar{f}_r$  as  $L_r = [W_r^1, W_r^2, \dots, W_r^p, a_r, \mathbf{V}_r]$  where  $W_r^p$  denotes the consumption of a resource  $p$  by partial route  $r$ . In our model, this corresponds to the total vehicle capacity consumed along the partial route (the only explicit resource considered).  $g_r$  is the number of unreachable nodes for the route  $r$ .  $\mathbf{V}_r$  is the vector of unreachable nodes, where  $V_r^i = 1$  if node  $i$  is unreachable for the route  $r$  and 0 otherwise.  $\bar{f}_r$  is the reduced cost of the partial route, computed after optimizing the departure time from the depot via Golden Section Search.

As shown by Taş et al. (2014a), the total service cost of a fixed route is a convex function of the vehicle's departure time from the depot. Therefore, for each partial route, we determine the optimal departure time from the depot via the Golden Section Search method (Chapra and Canale, 2011) and then compute the corresponding expected service cost and the reduced cost.

Let two partial routes  $r$  and  $r^*$  both reach the same node  $i$ , represented by labels  $(L_r, \bar{f}_r)$  and  $(L_{r^*}, \bar{f}_{r^*})$ . For the comparison, we need to compute the reduced cost of route  $r$  when its departure time is adjusted so that it arrives at the node  $i$  at the same time as route  $r^*$  ( $\bar{f}_{r2}$ ). This alignment is required to compare partial routes fairly. Route  $r$  dominates route  $r^*$  if the following conditions hold:

- i. Resource consumption: For every resource  $p$ :  $W_r^p \leq W_{r^*}^p$ .
- ii. Unreachable nodes (count):  $a_r \leq a_{r^*}$ .
- iii. Unreachable nodes (vector):  $V_r^b \leq V_{r^*}^b$ , for every node  $b$ .
- iv. Reduced cost (optimal departure time):  $\bar{f}_r \leq \bar{f}_{r^*}$ .
- v. Reduced cost (aligned arrival times):  $\bar{f}_{r2} \leq \bar{f}_{r^*}$ .

If all inequalities hold and at least one is strict, then route  $r$  dominates  $r^*$ , and label  $r^*$  can be safely discarded. These dominance rules ensure that only labels offering equal or better potential remain active. By eliminating dominated partial routes at each extension step, the computational complexity is significantly reduced while preserving the ability to generate improving columns. When no further extension is possible, completed routes are evaluated and feasible routes from depot back to depot with a negative reduced cost are then sent to the RLPMP.

#### 4.3.2. Acceleration Strategies

To accelerate the pricing subproblem, we first employ the ng-route relaxation technique (see, e.g., Baldacci et al., 2011), which has proven effective for the ESPPRC in CGH. This method allows controlled revisits outside a predefined neighborhood set. In this approach, revisits are not controlled

through an explicit visit limit but implicitly regulated by the size and composition of the ng-set associated with each delivery location. For a given location, the ng-set consists of a predefined number of its nearest neighboring locations, and locations belonging to this set are prohibited from being revisited within the same partial route. Re-visits are allowed only for the locations not included by these ng-sets, which provides a controlled relaxation of elementarity.

Since customers may have multiple delivery locations, this relaxation may lead to routes in which more than one location of the same customer appears during the pricing phase. In addition, the relaxation may also produce routes in which the same location node is visited more than once, whenever such revisits are permitted by the ng-set rule. This behavior is intentional and does not violate feasibility at this stage, as ng-routes are used solely as a search acceleration mechanism within the pricing subproblem. The feasibility of routes with respect to customer-level elementarity is ensured after route generation through a dedicated post-processing step. Specifically, once a candidate route with negative reduced cost is generated under the ng-route relaxation, it is post-processed before being added to the master problem: First, we restore customer-level elementarity by retaining only a single delivery location for each customer (the one corresponding to the first occurrence in the route representation) and removing all other occurrences of that customer's alternative locations. Second, if the ng-route relaxation produces repeated visits to the same location node (i.e., a location appears more than once in the route), we make the route elementary at the node level by keeping only the first occurrence of that location and removing subsequent repetitions. After this cleaning step, the total cost and reduced cost are recomputed using its optimal departure time (via the Golden Section Search). The route is added to the master problem only if its reduced cost remains negative, otherwise it is discarded.

The ng-sets employed in the relaxed pricing subproblem are dynamically updated across column generation iterations to progressively guide the search toward customer-level elementarity. This update mechanism is triggered based on the structure of candidate routes generated by the ng-route relaxation after post-processing. If a cleaned route remains improving (i.e., has negative reduced cost) and is added to the RLPMP, the algorithm records the fact that the corresponding customer was repeatedly visited in the relaxed (pre-cleaning) route. If, after post-processing, none of the candidate routes remains improving (i.e., all cleaned routes have non-negative reduced cost), the ng-sets are updated and the pricing subproblem is re-solved within the same column generation iteration. This restart ensures that the early-stopping strategy does not prematurely conclude that no improving column exists under the current dual values.

In the subsequent column generation iteration, all alternative locations of that customer are removed from the ng-sets of other locations. This update reduces the likelihood that multiple locations of the same customer will appear again in future ng-routes, thereby strengthening customer-level elementarity without enforcing it rigidly from the outset. Through this mechanism, the pricing subproblem gradually transitions from a more relaxed search space; useful for early exploration, to a more restrictive structure that aligns with the true problem definition. The second mechanism is an early-stopping criterion in the pricing search. Instead of exhaustively enumerating all routes with negative reduced cost in each pricing

iteration, the label-correcting algorithm is terminated once a predefined number of negative reduced-cost elementary routes has been identified. All such routes obtained so far are then sent to the RLPMP. The third mechanism is using a column pool, which is maintained to store distinct routes generated during pricing, so that previously explored (and ultimately non-improving) routes are not regenerated in subsequent calls of the pricing procedure. The pool stores complete routes generated by the pricing subproblem but not transferred to the RLPMP because their reduced costs are nonnegative. After solving the RLPMP, we first compute the reduced costs of the stored routes using the new optimal dual values. If any route in the pool has negative reduced cost, it is directly added to the RLPMP without resolving the pricing subproblem. This strategy enables the reuse of previously generated routes and reduces computational effort.

Lastly, the pricing algorithm is accelerated through parallel label extension. The label-correcting process is inherently parallelizable, since different partial routes (labels) can be extended independently. In the proposed implementation, active labels are distributed across multiple CPU cores, and their extensions are processed concurrently. Thread-safe data structures and synchronization mechanisms are used to insert newly generated labels into the global label pool and to prevent duplicate processing. This parallelization substantially reduces the computational burden of the pricing subproblem, particularly for larger instances where the number of labels grows rapidly.

#### **4.4. Termination and Integral Solution**

The CGH, consisting of solving RLPMP and pricing subproblem, continues until the pricing subproblem fails to find any elementary route with a negative reduced cost. At that point, optimality has been reached for the linear relaxation of the master problem, and the current RLPMP objective value constitutes a valid lower bound on the optimal integer solution. By this stage, the algorithm typically accumulates a large and diverse collection of feasible routes. Once no improving columns remain, an integer programming model is solved over all the columns generated and sent to the RLPMP, yielding an upper bound on the optimal solution of the SVRPFDD. The difference between the lower bound and the upper bound indicates the optimality gap. A complete and structured pseudocode representation of the entire CGH is provided in the subsequent algorithms 1-4.

### **5. Computational Results**

In this section, we present a comprehensive computational study with four objectives. First, we verify the effectiveness of the proposed solution approach on well-known benchmark instances by solving the classical VRPRDL and its extension, namely the VRP with Home and Roaming Delivery Locations (VRPHRDL). Second, we report computational results for the problem proposed in this paper (SVRPFDD) and its extension with home and delivery locations, demonstrating how the solution approach performs under travel time uncertainty. Third, we extend the SVRPFDD by considering SDLs and analyze the resulting operational trade-offs. Finally, we report sensitivity analyses and derive managerial insights on the trade-off between transportation-related costs and service-related penalties, highlighting how different weight settings and delivery structures shift the resulting routing decisions.

All experiments are conducted on a computer equipped with an Intel Core i7-8700 (3.2 GHz) CPU and 32 GB RAM. The RLPMP is solved using IBM ILOG CPLEX Optimization Studio (v20.1), and the overall solution approach is implemented in C# and compiled in Microsoft Visual Studio.

**Algorithm 1:** CGH for SVRPF

```

// Initial Solution Construction
InitialRouteSet  $\leftarrow \emptyset$ 
UnassignedCustomers  $\leftarrow$  Set of all customers
UnvisitedLocations  $\leftarrow$  Set of all delivery locations of all
customers
For each vehicle do
    RemainingCapacity  $\leftarrow$  VehicleCapacity
    CurrentNode  $\leftarrow$  Depot
    Start an empty Route
    While UnassignedCustomers is not empty do
        NextLocation  $\leftarrow$  FNFL(UnvisitedLocations,
        CurrentNode, RemainingCapacity) // Algorithm 2
        If NextLocation does not exist then
            break // no feasible extension; close this route
        End If
        Customer  $\leftarrow$  CustomerOf(NextLocation)
        Add NextLocation to Route
        Update RemainingCapacity
        // remove the customer and all of its locations
        Remove Customer from UnassignedCustomers
        CurrentNode  $\leftarrow$  NextLocation
    End While
    Add Route to InitialRouteSet
End for

// Column Generation Loop
Initialize the Restricted Linear Programming Master
Problem (RLPMP)
Solve RLPMP using CPLEX
Obtain the optimal dual values from the RLPMP
PricingFlag  $\leftarrow$  1 // run pricing subproblem at least once
While PricingFlag = 1 do
    PricingFlag  $\leftarrow$  0
    RouteSet  $\leftarrow \emptyset$ 
    // Solve the pricing subproblem
    RouteSet  $\leftarrow$  SolvePricingSubproblem() // label-
correcting algorithm 3
    If RouteSet  $\neq \emptyset$  then
        Add all routes in RouteSet to the RLPMP
        PricingFlag  $\leftarrow$  1
    End If
    If PricingFlag = 1 then
        Re-solve RMP with updated column set
        Update dual values from the new RLPMP solution
    End If
End While
// Final Integer Solution
If the RMP solution is not integer then
    Enforce binary (0–1) route-selection variables
    Solve the final master problem as an Integer Program
(IP) using CPLEX
End If
Return the selected set of routes

```

**Algorithm 2.** FNFL (UnvisitedLocations, CurrentNode, RemainingCapacity)

```

BestLocation  $\leftarrow$  null
BestDistance  $\leftarrow +\infty$ 
For each location p in UnvisitedLocations do
    Customer  $\leftarrow$  CustomerOf(p)
    If Demand(Customer)  $\leq$  RemainingCapacity then
        d  $\leftarrow$  Distance(CurrentNode, p)
        // Nearest-neighbor selection
        If d < BestDistance then
            BestDistance  $\leftarrow$  d
            BestLocation  $\leftarrow$  p
        End If
    End If
End For
Return BestLocation // null if none is feasible

```

**Algorithm 3.** Label-correcting Algorithm

```

Initialize LabelSet with an initial label at the depot
While there exists an extendable label do
    Select a Label  $\ell$  from LabelSet
    For each outgoing arc (i  $\rightarrow$  j) from  $\ell$  do
        (NewLabel, Success)  $\leftarrow$  ExtendLabel( $\ell$ , i  $\rightarrow$  j)
        // Algorithm 4
        If Success = 1 then
            Apply dominance rules to remove dominated labels
            Add NewLabel to LabelSet
        End If
    End For
    // For labels that return to depot
    For each Label  $\ell$  located at the depot do
        If ReducedCost( $\ell$ ) < 0 then
            Add the route corresponding to  $\ell$  to
            GeneratedRouteSet
        End If
    End For
End While
Return GeneratedRouteSet

```

**Algorithm 4.** ExtendLabel( $\ell$ , Arc)

```

If Arc leads to a node already visited in  $\ell$  then
    return (null, 0)
End If
If adding Arc violates vehicle capacity then
    return (null, 0)
End If
Construct NewLabel with:
- Updated route including the new arc
- Updated resource consumptions
- Updated reduced cost
- Updated set of visited nodes
- Updated set of visited alternative locations
Return (NewLabel, 1)

```

### 5.1. Performance of CGH on the Deterministic VRPRDL and VRPHRDL

We first validate the performance of the CGH under a setting with deterministic travel times and hard time window constraints. We focus on benchmark sets that comprise 60 instances: 40 base instances originally introduced by Reyes et al. (2017), complemented with two additional 10-instance variant sets (V1 and V2) proposed by Ozbaygin et al. (2017). Tilk et al. (2021) solves these instances both for the VRPRDL and the VRPHRDL and report the optimal or best-known solution values. Once the pricing problem can no longer identify improving routes, the column generation algorithm terminates with an optimal solution to the relaxed problem, and this solution corresponds to a valid lower bound (LB). An integer programming problem is then solved over all the routes sent to the RLPMP to generate a feasible route plan, leading to an upper bound (UB).

**Table 3.** Performance of the CGH on the deterministic VRPRDL instances

Instance	Opt	$m$	UB	GAP (%)	Time (s)	Instance	Opt	$m$	UB	GAP (%)	Time (s)
1	901	4	<b>901</b>	<b>0.00</b>	196.81	31	4935	18	5167	4.70	842.28
2	1286	5	<b>1286</b>	<b>0.00</b>	197.70	32	5258	19	5534	5.25	805.45
3	991	4	<b>991</b>	<b>0.00</b>	356.57	33	5061	18	5286	4.45	522.57
4	1062	5	<b>1062</b>	<b>0.00</b>	289.16	34	5218	18	5433	4.12	853.44
5	1832	6	<b>1832</b>	<b>0.00</b>	252.94	35	5498	20	5605	1.95	759.78
6	1294	5	<b>1294</b>	<b>0.00</b>	209.69	36	6498	22	6651	2.35	738.70
7	1155	4	<b>1155</b>	<b>0.00</b>	313.67	37	4830	18	4916	1.78	938.25
8	1455	6	<b>1455</b>	<b>0.00</b>	276.43	38	5604	20	5654	0.89	614.98
9	1260	5	<b>1260</b>	<b>0.00</b>	315.14	39	5841	21	6187	5.92	971.82
10	1684	7	<b>1684</b>	<b>0.00</b>	283.42	40	4995	19	5158	3.26	1252.05
11	1922	7	<b>1922</b>	<b>0.00</b>	281.48	41_v1	3203	10	3254	1.59	483.35
12	2324	8	2328	0.17	297.74	42_v1	2799	9	2804	0.18	496.17
13	1747	6	1754	0.40	277.04	43_v1	2603	8	2647	1.69	363.53
14	1273	6	<b>1273</b>	<b>0.00</b>	468.02	44_v1	2261	7	2262	0.04	365.09
15	1694	6	1696	0.12	323.78	45_v1	3217	10	3225	0.25	564.15
16	1938	7	1940	0.10	346.93	46_v1	2805	9	2806	0.04	658.10
17	1965	8	1968	0.15	544.81	47_v1	3339	10	3344	0.15	531.10
18	1827	7	<b>1827</b>	<b>0.00</b>	310.74	48_v1	3325	10	3329	0.12	320.98
19	2083	7	2084	0.05	479.30	49_v1	3534	11	3539	0.14	474.22
20	1822	6	1823	0.05	518.37	50_v1	2752	9	2756	0.15	370.69
21	3761	13	3784	0.61	659.12	41_v2	2133	7	2151	0.84	376.81
22	2828	10	2829	0.04	580.90	42_v2	1946	7	2006	3.08	362.78
23	4440	16	4465	0.56	705.84	43_v2	1966	8	1982	0.81	364.55
24	3378	11	3426	1.42	521.23	44_v2	1610	5	1616	0.37	363.22
25	3161	11	3194	1.04	295.57	45_v2	2478	8	2500	0.89	364.97
26	4536	16	<b>4536</b>	<b>0.00</b>	272.23	46_v2	2469	8	2504	1.42	371.06
27	2865	10	2881	0.56	536.44	47_v2	1946	7	1952	0.31	375.99
28	4173	14	4174	0.02	720.28	48_v2	2380	8	2381	0.04	362.77
29	3964	15	3995	0.78	803.54	49_v2	2492	8	2529	1.48	360.37
30	4107	14	4154	1.14	491.05	50_v2	2443	8	2470	1.11	415.98

Avg. Opt/BKS: 2902.78

Avg.  $m$ : 10.15

Avg. UB: 2943.68

Avg. GAP (%): 0.94

Avg. Time (s): 479.52

The results obtained by the proposed solution procedure for the VRPRDL and VRPHRDL are presented in Table 3 and Table 4, respectively. In both tables, the columns “Instance” and “Opt/BKS”, report the instance name and the reference solution value (the optimal or best-known) reported by Tilk et al. (2021) and Ozbaygin et al. (2017). Note that in these tables, non-italic values represent optimal solutions,

whereas *italic* values correspond to solutions whose optimality has not been proven. The remaining columns present the performance of the proposed CGH. Specifically, “*m*” denotes the number of vehicles used, “UB” gives the objective value of the integer solution obtained by CGH, “GAP (%)” reports the percentage deviation of the CGH solution (UB) from the corresponding Opt/BKS value, and “Time (s)” provides the runtime in seconds. Note that when the obtained UB equals to the optimal or better than best-known value, it is presented in bold.

Across the 60 VRPRDL instances, CGH achieves an average gap of 0.94% (maximum 5.92%) with an average runtime of 479.5 seconds. CGH matches the optimal solutions exactly in 14 out of 60 instances, and remains very close to the optimal/BKS in most cases: 70.0% of instances are within 1%, 86.7% within 2%, and 96.7% within 5%. The gaps are higher in the large-sized instances which are more difficult to solve (instances 30-39).

**Table 4.** Performance of the CGH on the deterministic VRPHRDL instances

Instance	Opt/BKS	<i>m</i>	UB	GAP (%)	Time (s)	Instance	Opt/BKS	<i>m</i>	UB	GAP (%)	Time (s)
1	773	3	<b>773</b>	<b>0.00</b>	227.36	31	3666	12	3779	3.08	852.93
2	1065	4	<b>1065</b>	<b>0.00</b>	215.65	32	3958	12	3993	0.88	815.03
3	988	3	<b>988</b>	<b>0.00</b>	368.40	33	<i>3630</i>	12	<b>3622</b>	<b>-0.22</b>	548.66
4	914	3	<b>914</b>	<b>0.00</b>	306.19	34	3694	13	3747	1.43	870.82
5	1710	6	<b>1710</b>	<b>0.00</b>	286.76	35	3255	10	<b>3207</b>	<b>-1.47</b>	783.83
6	1099	4	<b>1099</b>	<b>0.00</b>	246.51	36	<i>4525</i>	13	<b>4425</b>	<b>-2.21</b>	764.72
7	996	3	<b>996</b>	<b>0.00</b>	317.96	37	<i>3217</i>	11	3301	2.61	947.21
8	1346	5	<b>1346</b>	<b>0.00</b>	296.85	38	<i>3976</i>	12	<b>3954</b>	<b>-0.55</b>	624.19
9	997	4	<b>997</b>	<b>0.00</b>	331.28	39	<i>4316</i>	13	4388	1.67	984.81
10	1166	4	<b>1166</b>	<b>0.00</b>	292.08	40	<i>3680</i>	11	<b>3587</b>	<b>-2.53</b>	1279.86
11	1587	5	1599	0.76	299.10	41_v1	2662	8	<b>2662</b>	<b>0.00</b>	530.73
12	1808	6	1819	0.61	300.09	42_v1	2610	8	2616	0.23	519.85
13	1563	6	<b>1563</b>	<b>0.00</b>	313.60	43_v1	2260	7	2314	2.39	428.25
14	1058	4	<b>1058</b>	<b>0.00</b>	489.50	44_v1	2147	7	<b>2147</b>	<b>0.00</b>	419.14
15	1347	5	1355	0.59	355.32	45_v1	3172	10	<b>3172</b>	<b>0.00</b>	625.76
16	1517	5	<b>1517</b>	<b>0.00</b>	375.67	46_v1	2616	8	<b>2616</b>	<b>0.00</b>	698.89
17	1445	5	<b>1445</b>	<b>0.00</b>	565.80	47_v1	3010	9	<b>3010</b>	<b>0.00</b>	560.93
18	1627	5	1631	0.25	337.02	48_v1	3278	10	3285	0.21	394.16
19	1461	5	<b>1461</b>	<b>0.00</b>	483.53	49_v1	3514	11	3517	0.09	531.02
20	1715	6	<b>1715</b>	<b>0.00</b>	556.53	50_v1	2727	9	2732	0.18	430.43
21	2580	8	2644	2.48	669.90	41_v2	1998	7	2076	3.90	432.49
22	2206	7	2213	0.32	596.96	42_v2	1927	7	1961	1.76	420.94
23	3363	11	3526	4.85	721.46	43_v2	1830	6	1832	0.11	421.92
24	2569	8	2584	0.58	550.21	44_v2	1478	5	1480	0.14	401.13
25	2378	8	2395	0.71	310.14	45_v2	2466	8	2485	0.77	410.80
26	2845	9	<b>2845</b>	<b>0.00</b>	290.55	46_v2	2388	8	2411	0.96	431.44
27	2518	8	2529	0.44	553.03	47_v2	1848	7	1868	1.08	432.08
28	2758	8	2784	0.94	749.87	48_v2	2264	7	2272	0.35	420.58
29	2892	9	2910	0.62	816.54	49_v2	2457	8	<b>2457</b>	<b>0.00</b>	405.74
30	2691	8	2702	0.41	526.01	50_v2	2302	8	2374	3.13	491.64

Avg. Opt/BKS: 2330.88

Avg. *m*: 7.53

Avg. UB: 2343.98

Avg. GAP (%): 0.53

Avg. Time (s): 510.50

For the VRPHRDL, CGH achieves an average gap of 0.53% (maximum 4.85%) with an average runtime of 510.5 seconds. The proposed approach obtains the optimal/BKS for 23 out of 60 instances (38.3%). More importantly, the CGH improves the current BKS reported in the literature for five instances (33, 35, 36, 38, and 40). Furthermore, all instances are within 5% of the optimal/BKS.

Compared to the VRPRDL, the slightly higher average runtime is consistent with larger feasibility structure created by “*home and roaming*” setting. It is observed that the solution quality remains strong and stable, confirming that the CGH is effective even under the added flexibility of the VRPHRDL. Overall, these results provide a strong validation of the proposed solution approach where it reproduces a nontrivial portion of optimal/best-known solutions and stays consistently close to the BKS across all 60 instances. It further maintains computation times on the order of minutes per instance, confirming efficiency on standard deterministic benchmarks.

## 5.2. Performance of CGH on SVRPFDD Benchmarks

In this section, we examine the performance of the solution approach considering the SVRPFDD. We report, for each instance, the number of vehicles used in the final integer solution, the LB, the UB, the gap between the LB and the UB, and the total computational time. All stochastic experiments use the same parameter setting aligned with the soft time-window cost structure used in Taş et al. (2013, 2014a). In the deterministic experiments (Section 5.1), we follow the original benchmark setting and use a large vehicle capacity ( $Q = 750$ ). Since the capacity is the only resource considered in the pricing subproblem and the original value makes the problem nearly uncapacitated, we set the vehicle capacity to 150 in the SVRPFDD experiments. Unlike the original benchmark data, which do not explicitly report service times at customer locations, we incorporate a constant service time (i.e.,  $b_i=15$ ) for each location in the SVRPFDD. This addition reflects a practical feature of last-mile operations, each stop requires handling and handover time. The weighted-cost coefficient,  $\mu$  is set to 0.5, and the driver shift length,  $w$  is set to 480. The values of the travel and penalty coefficients are given as follows:  $c_t=1$ ,  $c_f = 400$ ,  $c_e = 0.1$ ,  $c_l = 1$ , and  $c_o = 5/6$ . Stochastic travel times are modeled via Gamma distributions with shape and scale parameters equal to 1, and the expected earliness/lateness components are computed adopting the analytical framework proposed in Taş et al. (2013). This framework ensures consistency with established soft time-window routing practice under travel-time uncertainty.

The pricing problem is solved by a label-correcting ESPPRC variant under ng-route relaxation, and we explicitly define ng-sets over delivery-location nodes to control admissible revisits in the presence of multiple alternative locations per customer. Concretely, for each location node, we construct the corresponding ng-set as the set of its  $p$  closest location nodes, where  $p$  is equal to  $2|C|$ . In each pricing iteration, we use an early-stopping rule that terminates the label search once 10 elementary routes with negative reduced cost have been identified. Table 5 summarizes solutions obtained by the CGH for the SVRPFDD. Over the full test set (60 instances), the average gap between the UB and LB is 5.87%, with gaps ranging from 1.57% to 15.13%. Computational times are moderate overall: the average runtime is 2940.32 seconds, with a minimum of 207.78 seconds and a maximum of 13920.17 seconds. The number of routes selected in the final integer solution varies substantially within the instance size, ranging from 5 to 43, with an average of 17.63 routes. As concrete examples from Table 5, small instances show tight gap, whereas the larger instances exhibit wider gaps, reflecting the increased difficulty of closing the integrality gap under stochastic penalties as the route set grows.

**Table 5.** CGH results for the stochastic SVRPFD

Instance	$m$	LB	UB	GAP (%)	Time (s)	Instance	$m$	LB	UB	GAP (%)	Time (s)
1	6	1565.52	1700.05	7.91	258.40	31	41	10899.01	12307.42	11.44	3467.55
2	5	1486.82	1541.08	3.52	629.09	32	41	11376.62	12676.28	10.25	11307.28
3	5	1482.34	1537.25	3.57	247.54	33	43	11193.54	12824.99	12.72	13920.17
4	5	1481.31	1524.41	2.83	207.78	34	39	10571.72	12366.3	14.51	11778.57
5	6	1990.48	2025.69	1.74	566.70	35	41	10896.09	11875.97	8.25	12207.37
6	7	2110.44	2144.03	1.57	565.69	36	41	11367.53	13133.02	13.44	13645.77
7	7	2033.37	2111.26	3.69	577.09	37	40	10985.19	12647.91	13.15	12467.02
8	7	2008.6	2145.02	6.36	567.59	38	39	10907.6	12442.85	12.34	12462.04
9	7	2064.58	2122.14	2.71	458.40	39	40	10806.64	12617.55	14.35	11211.79
10	8	2254.11	2393.62	5.83	426.32	40	38	10225.72	12048.51	15.13	12100
11	10	3095.76	3235.21	4.31	550.45	41_v1	15	4658.74	4841.39	3.77	1067.1
12	9	2706.15	2860.31	5.39	675.3	42_v1	14	4402.98	4527.26	2.75	1409.07
13	11	3120.48	3265.48	4.44	419.77	43_v1	14	4390.77	4511.48	2.68	1157.49
14	10	2825.82	2911.48	2.94	524.89	44_v1	14	4267.06	4427	3.61	1368.06
15	11	3253.13	3352.65	2.97	611.16	45_v1	15	4509.53	4678.09	3.60	1287.16
16	10	3036.23	3123.09	2.78	589.25	46_v1	14	4351.27	4508.26	3.48	1102.1
17	10	3098.63	3186.95	2.77	586.11	47_v1	14	4545.63	4691.75	3.11	1386.89
18	12	3633.79	3734.81	2.70	608.94	48_v1	14	4487.46	4640.59	3.30	1224.46
19	10	2984.20	3098.64	3.69	629.32	49_v1	14	4592.16	4709.75	2.50	1386.23
20	10	2955.78	3045.81	2.96	456.75	50_v1	15	4692.75	4874.51	3.73	1160.58
21	21	5754.24	6189.27	7.03	1548.83	41_v2	15	4386.52	4591.65	4.47	1344.9
22	21	5671.51	6243.49	9.16	1846.75	42_v2	14	4122.31	4254.06	3.10	1347.04
23	21	5956.72	6492.95	8.26	1702.42	43_v2	14	4024.62	4147.3	2.96	1256.16
24	21	5958.09	6507.16	8.44	1827.53	44_v2	14	4024.24	4212.43	4.47	1336.26
25	20	5685.31	6210.94	8.46	1810.34	45_v2	15	4126.94	4305.85	4.16	1342.68
26	21	5979.07	6534.02	8.49	3560.52	46_v2	14	4009.79	4193.48	4.38	1170.62
27	20	5740.73	6279.02	8.57	3576.88	47_v2	14	4257.99	4394.14	3.10	1299.59
28	18	5036.20	5453.9	7.66	3628.97	48_v2	14	4169.74	4327.36	3.64	1423.41
29	20	5671.44	6239.82	9.11	3613.3	49_v2	14	4209.57	4320	2.56	1118.45
30	20	5766.11	6279.46	8.18	3185.31	50_v2	15	4459.86	4605.53	3.16	1206.26

Avg.  $m$ : 17.63

Avg. LB: 5038.77

Avg. UB: 5469.86

Avg. GAP (%): 5.87

Avg. Time (s): 2940.32

Table 6 reports the results obtained for the SVRPFD with home delivery case using the same algorithmic configuration and coefficients. Across the same 60 instances, the average gap between the UB and the LB decreases to 5.55%, with a minimum of 1.20% and a maximum of 13.54%. The average runtime increases to 3061.45 seconds (minimum 211.26 seconds, maximum 16667.93 seconds), while the average number of routes in the final solution remains essentially unchanged (17.60). Hence, introducing home delivery does not systematically change the number of vehicles used, but it does change route composition and expected penalty exposure, which is reflected in both improved upper bounds and slightly tighter gaps on average.

A direct comparison of these tables shows that enabling home delivery during all day yields a consistent reduction in UB across almost all instances. Specifically, the UB is strictly better in 59 out of 60 instances and equal in the remaining one (instance 42\_v2, where UB remains 4147.30). Averaged over all instances, the UB decreases by 52.6 cost units, corresponding to an average improvement of approximately 0.77% relative to the no-home baseline. This improvement comes mainly from lower deviations from time window bounds. In particular, the average total distance slightly changes, decreasing from 3729.78 to 3723.12 on average. In contrast, expected earliness decreases substantially

from 1197.82 to 540.74, and expected lateness decreases from 39.01 to 13.24. Expected overtime also slightly decreases, from 5.19 to 4.92. These results show that the main benefit of allowing home delivery during day performs better in terms of service, especially through a large reduction in expected earliness, while the route structure itself changes only very little. From a computational perspective, the average runtime increased by 4.12% compared to the no-home setting. Overall, the results indicate that CGH achieves competitive solution quality with respect to the available LBs, as reflected by the consistently small gap between UB and LB reported in Table 5 and Table 6, while requiring only moderate computational effort. They also demonstrate that the home-delivery extension yields a measurable and broadly consistent benefit in expected total cost under travel-time uncertainty.

**Table 6.** CGH results for the SVRPF instances with home-delivery variant

Instance	$m$	LB	UB	GAP (%)	Time (s)	Instance	$m$	LB	UB	GAP (%)	Time (s)
1	6	1548.05	1691.5	8.48	260.90	31	41	10883.70	12178.34	10.63	3511.30
2	5	1469.24	1538.61	4.51	636.28	32	41	11363.96	12587.63	9.72	11448.28
3	5	1465.49	1532.36	4.36	251.39	33	43	11183.24	12632.14	11.47	15107.05
4	5	1469.24	1522.88	3.52	211.26	34	39	10552.75	12047.17	12.40	11925.43
5	6	1971.06	2010.96	1.98	573.73	35	40	10875.93	11752.54	7.46	12360.08
6	7	2092.48	2141.74	2.30	572.47	36	41	11357.43	13010.63	12.71	13816.39
7	7	2019.62	2098.29	3.75	583.52	37	40	10967.56	12480.88	12.13	12622.67
8	7	1997.61	2140.05	6.66	575.24	38	39	10892.73	12285.29	11.34	16667.93
9	7	2052.43	2095.23	2.04	464.11	39	40	10786.19	12474.79	13.54	11352.28
10	8	2234.17	2374.78	5.92	431.56	40	38	10207.74	11674.83	12.57	12251.91
11	10	3079.53	3200.44	3.78	556.76	41_v1	15	4637.78	4828.35	3.95	1080.33
12	9	2692.82	2814.55	4.33	683.60	42_v1	14	4382.76	4511.37	2.85	1425.04
13	11	3107.77	3227.26	3.70	426.09	43_v1	14	4372.72	4507.03	2.98	1171.97
14	10	2812.83	2900.32	3.02	531.27	44_v1	14	4247.85	4409	3.66	1385.99
15	11	3235.26	3299.13	1.94	618.94	45_v1	15	4494.96	4667.73	3.70	1302.85
16	10	3019.28	3068.44	1.60	595.81	46_v1	14	4332.57	4471.89	3.12	1116.13
17	10	3087.15	3124.61	1.20	593.33	47_v1	14	4526.10	4644.66	2.55	1404.60
18	12	3613.60	3713.74	2.70	616.15	48_v1	14	4476.88	4611.14	2.91	1239.89
19	10	2967.01	3077.22	3.58	636.69	49_v1	14	4576.11	4692.22	2.47	1403.67
20	10	2944.08	3041.29	3.20	462.87	50_v1	15	4673.93	4859.93	3.83	1175.33
21	20	5738.66	6114.12	6.14	1568.21	41_v2	15	4370.45	4585.4	4.69	1362.25
22	21	5655.51	6181.74	8.51	1869.00	42_v2	14	4105.34	4244.1	3.27	1364.60
23	21	5939.63	6445.88	7.85	1723.04	43_v2	14	4012.38	4147.3	3.25	1271.31
24	21	5944.00	6437.88	7.67	1850.63	44_v2	14	4005.57	4201.97	4.67	1353.71
25	20	5672.62	6131.08	7.48	1833.43	45_v2	15	4114.73	4302.81	4.37	1359.91
26	21	5962.50	6482.09	8.02	3605.37	46_v2	14	3993.14	4176.6	4.39	1185.22
27	20	5729.85	6209.28	7.72	3622.56	47_v2	14	4242.64	4388.15	3.32	1315.50
28	18	5020.64	5382.52	6.72	3675.01	48_v2	14	4152.81	4324.95	3.98	1440.81
29	20	5658.99	6183.19	8.48	3657.70	49_v2	14	4196.15	4315.07	2.76	1132.04
30	20	5752.26	6246.17	7.91	3225.21	50_v2	15	4449.29	4594.56	3.16	1220.57

Avg.  $m$ : 17.60

Avg. LB: 5023.15

Avg. UB: 5417.26

Avg. GAP (%): 5.55

Avg. Time (s): 3061.45

### 5.3. Results for the SVRPF with Shared Delivery Locations

We extend the stochastic SVRPF by allowing a subset of customers to be consolidated at pre-defined SDL nodes. Note that all SDL nodes are assumed to have a wide service window  $[0,720]$ . The key prerequisite for using SDLs effectively is to identify which customers are eligible and assign each eligible customer to at most one SDL in a consistent and capacity-aware manner. Since the SDL nodes and their customer assignments are predetermined in our experiments, we first construct a feasible

customer-SDL assignment and then solve the resulting SVRPFD-SDL problem. The overall goal is to exploit the location/capacity advantages of SDLs while preserving operational feasibility.

We employ a two-stage procedure with (i) candidate filtering and (ii) scoring. In the first stage, we exclude the customers who have only one delivery location and are feasible throughout the full horizon  $[0,720]$ . More specifically, these customers are assumed to prefer home delivery and are therefore not diverted to SDLs. Then, for each SDL  $o$  all eligible customer-location pairs  $(c, i)$  are evaluated using a distance-demand score:  $Score_{c,i}(o) = p_1 \left( \frac{d_{io}}{Max_d(o)} \right)^2 + p_2 \left( \frac{q_c}{Max_q(o)} \right)^2$ , where  $p_1 + p_2 = 1$ . In this equation,  $d_{io}$  is the distance between customer  $c$ 's delivery location  $i$  and SDL  $o$ , and  $q_c$  is the customer demand. Moreover,  $Max_d(o)$  and  $Max_q(o)$  denote the maximum distance and the maximum demand among all the eligible candidates considered for SDL  $o$ , respectively. These maxima are computed separately for each SDL, so that the two criteria are normalized on comparable scales. Additionally, the squared terms intentionally penalize large distances and large demands more strongly, thereby reducing the likelihood of assigning far or high-demand customers to an SDL. Candidates are sorted by increasing score (lower is better), and each SDL selects a fixed number of  $m$  customers, subject to the constraint that each customer can be assigned to at most one SDL. Note that each customer may have at most one SDL in the set of alternative delivery locations. Therefore, when a customer is assigned to an SDL within this procedure, it is excluded from the scoring phase for all other SDLs.

In addition to the parameter indicating the number of customers assigned per SDL ( $m$ ), the SVRPFD-SDL considers one more parameter, which is the number of SDLs ( $k$ ). In our experiments, we set  $k$  and  $m$  by instance size as follows: for 15-customer instances  $k = 2, m = 2$ ; for 20- and 30-customer instances  $k = 3, m = 2$ ; for 40- and 60-customer instances  $k = 4, m = 3$ ; and for 120-customer instances  $k = 6, m = 3$ .

To examine the sensitivity of the SDL customer-assignment scoring weights, we vary  $p_1$  while keeping  $p_2 = 1 - p_1$ . Since evaluating all weight settings on the full benchmark set would be computationally expensive, this test is conducted on a representative subset of instances spanning different sizes and structural variants. Specifically, the subset includes instances 1, 3, 7, 20, 15, 17, 21, 29, 36, 40, and as well as two cases from the V1 variant set (41 and 48) and two cases from the V2 variant set (43 and 47). Table 7 reports the resulting upper bounds (UB) and relative improvements over the no-SDL baseline (original problem) for each  $p_1$  value. Overall, smaller values of  $p_1$  (i.e., placing relatively more emphasis on demand through  $p_2$ ) yield more consistent improvements across this subset, whereas larger  $p_1$  values, which prioritize proximity more strongly tend to reduce or eliminate the gains in several instances. Based on this analysis, we fix  $p_1=0.2$  (and  $p_2=0.8$ ) for the full SDL experiments.

Using the selected SDL assignment policy, Table 8 reports the UB values of the baseline stochastic setting (“Original Problem”) and the SDL extension (“SDL Case”) for the full set of 60 instances. On average, the SVRPFD-SDL reduces the UB from 5469.86 to 5449.56, corresponding to an average improvement of 0.69%. Across the 60 instances, the SDL case yields a strict UB decrease in 40 instances, while 20 instances remain unchanged. This indicates that SDLs can provide measurable, but

not universal savings, with most gains concentrated in specific instance families. The highest improvements occur in small- and medium-sized instances, where consolidation opportunities and route restructuring effects are more pronounced. For example, improvements of 5.37% (instance 5), 4.83% (instance 6), 3.92% (instance 7), and 3.01% (instance 8) show that SDL visits can substantially reduce the expected cost where several customers can be served via a well-placed SDL without overly fragmenting routes. For the larger instances (e.g., instances 30-39 with 120 customers), SDL case produces essentially no change. The reason behind this observation is as follow: as instance size grows, the fleet size typically increases and routes are already heavily shaped by capacity considerations, so consolidating a small subset of customers at SDLs does not materially alter the global route plan. Moreover, SDL assignments concentrate multiple customers into the shared nodes, effectively creating macro-stops. Although SDL locations may be geographically convenient, combining several customers at one shared point can make route planning more difficult. In some cases, the SDL location does not fit well with the sequence of nearby customers, which may disturb the natural route flow and limit the benefit of consolidation. In such cases, serving an SDL may need to be visited separately rather than being integrated with nearby non-SDL stops; this can offset potential distance savings and result in negligible net improvement. Over the V1 group, SDL provides an average improvement of approximately 0.63%, and for the V2 group the average improvement is about 0.57%, confirming that the benefit is not restricted to a single structural pattern. However, these improvements are generally modest (often below 1%) except for a few instances where consolidation aligns particularly well with the geometry and demand distribution.

**Table 7.** Sensitivity of SDL customer-assignment weights ( $p_1, p_2$ )

Instance	Original Problem	$p_1 = 0.2$		$p_1 = 0.4$		$p_1 = 0.6$		$p_1 = 0.8$		$p_1 = 1.0$	
	UB	UB	Imp. %	UB	Imp. %	UB	Imp. %	UB	Imp. %	UB	Imp. %
1	1700.05	1685.21	0.87	1681.90	1.07	1698.80	0.07	1696.89	0.19	1699.74	0.02
3	1537.25	1526.23	0.72	1526.92	0.67	1526.23	0.72	1526.23	0.72	1517.23	1.30
7	2111.26	2028.56	3.92	2026.98	3.99	2025.41	4.07	2073.46	1.79	2091.15	0.95
10	2393.62	2393.62	0.00	2393.62	0.00	2344.72	2.04	2344.72	2.04	2390.77	0.12
15	3352.65	3265.27	2.61	3264.56	2.63	3264.58	2.63	3303.24	1.47	3345.35	0.22
17	3186.95	3143.13	1.37	3186.95	0.00	3186.95	0.00	3186.89	0.00	3186.95	0.00
21	6189.27	6189.27	0.00	6188.73	0.01	6188.73	0.01	6188.73	0.01	6188.73	0.01
29	6239.82	6239.82	0.00	6213.32	0.42	6239.82	0.00	6239.82	0.00	6239.82	0.00
36	13133.02	13133.02	0.00	13133.02	0.00	13051.18	0.62	13133.02	0.00	13119.64	0.10
40	12048.51	12048.51	0.00	12048.51	0.00	12048.51	0.00	12048.51	0.00	12048.51	0.00
42_v1	4527.26	4499.79	0.61	4518.27	0.20	4519.58	0.17	4526.43	0.02	4524.69	0.06
49_v1	4709.75	4706.27	0.07	4684.00	0.55	4708.06	0.04	4653.89	1.19	4704.22	0.12
44_v2	4212.43	4118.99	2.22	4124.41	2.09	4192.72	0.47	4109.10	2.45	4164.25	1.14
48_v2	4327.36	4322.43	0.11	4323.30	0.09	4322.46	0.11	4327.00	0.01	4323.95	0.08
Average	4976.37	<b>4950.01</b>	<b>0.89</b>	4951.04	0.84	4951.27	0.78	4954.14	0.71	4967.50	0.29

These results are consistent with the sensitivity results: as  $p_1$  increases and the assignment becomes more distance-driven, the procedure may select customers that are close to an SDL in isolation but less compatible with the global route structure, further weakening the integrability of SDL visits, an effect

that becomes more evident in large instances. Taken together, the results in Table 8 suggest that the SDL extension is most beneficial when SDL assignments can (a) substitute multiple dispersed customer visits by a single consolidated stop without creating capacity-driven route fragmentation, and (b) keep the assigned SDL customer set “balanced” in demand (which is exactly what the  $p_1 = 0.2$  choice encourages).

**Table 8.** CGH results for the SVRPFD-SDL instances

Instance	Original Problem		SDL Case		Instance	Original Problem		SDL Case	
	UB		UB	Imp. %		UB		UB	Imp. %
1	1700.05		1685.21	0.87	31	12307.42		12307.42	0.00
2	1541.08		1520.16	1.36	32	12676.28		12676.28	0.00
3	1537.25		1526.23	0.72	33	12824.99		12824.99	0.00
4	1524.41		1509.04	1.01	34	12366.3		12366.30	0.00
5	2025.69		1916.97	5.37	35	11875.97		11875.54	0.00
6	2144.03		2040.42	4.83	36	13133.02		13133.02	0.00
7	2111.26		2028.56	3.92	37	12647.91		12647.41	0.00
8	2145.02		2080.53	3.01	38	12442.85		12442.85	0.00
9	2122.14		2088.02	1.61	39	12617.55		12617.58	0.00
10	2393.62		2393.62	0.00	40	12048.51		12048.51	0.00
11	3235.21		3188.66	1.44	41_v1	4841.39		4840.85	0.01
12	2860.31		2853.10	0.25	42_v1	4527.26		4499.79	0.61
13	3265.48		3265.48	0.00	43_v1	4511.48		4497.57	0.31
14	2911.48		2906.27	0.18	44_v1	4427		4330.42	2.18
15	3352.65		3265.27	2.61	45_v1	4678.09		4678.09	0.00
16	3123.09		3119.42	0.12	46_v1	4508.26		4434.97	1.63
17	3186.95		3143.13	1.37	47_v1	4691.75		4648.27	0.93
18	3734.81		3711.36	0.63	48_v1	4640.59		4634.98	0.12
19	3098.64		3097.62	0.03	49_v1	4709.75		4706.27	0.07
20	3045.81		3045.81	0.00	50_v1	4874.51		4851.52	0.47
21	6189.27		6189.27	0.00	41_v2	4591.65		4570.35	0.46
22	6243.49		6235.05	0.14	42_v2	4254.06		4208.39	1.07
23	6492.95		6489.77	0.05	43_v2	4147.3		4126.52	0.50
24	6507.16		6506.43	0.01	44_v2	4212.43		4118.99	2.22
25	6210.94		6210.94	0.00	45_v2	4305.85		4301.81	0.09
26	6534.02		6534.02	0.00	46_v2	4193.48		4167.92	0.61
27	6279.02		6279.15	0.00	47_v2	4394.14		4388.09	0.14
28	5453.9		5453.90	0.00	48_v2	4327.36		4322.43	0.11
29	6239.82		6239.82	0.00	49_v2	4320		4308.58	0.26
30	6279.46		6279.46	0.00	50_v2	4605.53		4595.45	0.22

Avg. UB for SVRPFD: 5469.86

Avg. UB for SVRPFD-SDL: 5449.56

Avg. Imp %: 0.69

#### 5.4. Sensitivity Analyses

To analyze the sensitivity of the proposed SVRPFD and SVRPFD-SDL formulations, we conducted three complementary analyses. The first experiment varies the objective-function weight  $\mu$ , which controls the balance between transportation costs and service costs. The second experiment increases the coefficient of variation (CV) of stochastic travel times from 1 to 4 to test how a more uncertain operating environment changes routing behavior. The third experiment studies an alternative SDL coordination generation policy derived from already-observed no-SDL routes. Together, these analyses clarify how the model reacts to different managerial priorities, different uncertainty levels, and different ways of constructing delivery flexibility. The experiments are conducted on the same representative subset of instances used in Section 5.3, spanning different sizes and structural variants. Specifically, the

subset includes instances 1, 3, 7, 20, 15, 17, 21, 29, 36, and 40, as well as two cases from the V1 variant set (41 and 48) and two cases from the V2 variant set (43 and 47).

#### 5.4.1. Effects of the Objective-Function Weight

In the proposed objective (1),  $\mu$  determines the relative emphasis assigned to the transportation cost component versus the service cost component. The baseline setting is  $\mu = 0.5$ . We additionally tested by setting  $\mu$  to 0.2 and 0.8 to represent a stronger service-oriented and transportation-oriented perspective, respectively. Table 9 reports the average performance metrics across tested instances, and the average effects are summarized more explicitly in Table 10 by comparing each alternative  $\mu$  value against the baseline. When  $\mu$  is reduced to 0.2, the model gives much more importance to service quality. Accordingly, the average service cost decreases sharply from 171.95 to 60.72 (a reduction of 64.69%), while average earliness and lateness also drop considerably. This improvement is obtained at the expense of a modest increase in transportation cost (an increase of 1.60%), slightly longer routes, and a marked increase in overtime because vehicles accept more operational effort to protect service quality.

In contrast, when  $\mu$  is increased to 0.8, the model becomes more transportation-driven and the average transportation cost decreases slightly from 9780.79 to 9743.22 (a decrease of 0.38%), but this is accompanied by a clear deterioration in service performance, with the expected service cost rising to 239.23 (an increase of 39.13%), the expected earliness rising to 1617.37 (an increase of 36.03%), and the expected lateness rising to 77.50 (an increase of 48.30%). Therefore,  $\mu$  directly controls whether the routing plan prioritizes operational efficiency or customer-service quality. The baseline  $\mu = 0.5$  provides a balanced compromise, while  $\mu = 0.2$  and  $\mu = 0.8$  reveal the two extreme managerial attitudes.

**Table 9.** The average performance of different  $\mu$  across tested instances

Case	#Routes	Total Distance	Expected Earliness	Expected Lateness	Expected Overtime	Transportation Cost	Service Cost
Baseline ( $\mu = 0.5$ )	15.86	3434.86	1189.00	52.26	3.69	9780.79	171.95
$\mu = 0.2$	15.93	3559.29	482.17	12.51	7.91	9937.31	60.72
$\mu = 0.8$	15.86	3397.29	1617.37	77.50	3.69	9743.22	239.23

**Table 10.** The average percentage change of different  $\mu$  across tested instances

Case	Transportation Cost	Service Cost	Expected Earliness	Expected Lateness	Expected Overtime
$\mu = 0.2$ vs $\mu = 0.5$	↑ 1.60%	↓ 64.69%	↓ 59.45%	↓ 76.06%	↑ 114.36%
$\mu = 0.8$ vs $\mu = 0.5$	↓ 0.38%	↑ 39.13%	↑ 36.03%	↑ 48.30%	→ 0.00%

#### 5.4.2. Effects of the Coefficient of Variation

The second experiment investigates the impact of stronger travel-time uncertainty on solution quality. In the baseline setting, the coefficient of variation is set to 1. This parameter is then increased to 4, while all other model components and parameter settings are kept unchanged. The average results are reported in Table 11, while the corresponding average comparison and percentage changes are summarized in Table 12.

**Table 11.** Average results for CV = 1 and CV = 4

Case	#Routes	Total Distance	Expected Earliness	Expected Lateness	Expected Overtime	Transportation Cost	Service Cost
Baseline (CV = 1)	15.93	3434.86	1189.00	52.26	3.69	4890.40	85.97
High variability (CV = 4)	15.86	3449.79	1685.11	91.52	37.73	4912.04	130.01

**Table 12.** Average percentage changes under different CV levels.

Comparison	Transportation Cost	Service Cost	Earliness	Lateness	Overtime
CV = 4 vs CV = 1	↑ 0.44%	↑ 51.23%	↑ 41.72%	↑ 75.12%	↑ 922.49%

According to the results, increasing the coefficient of variation from 1 to 4 substantially amplifies the effect of uncertainty on service reliability, while having only a limited effect on the routing structure itself. On average, the number of routes changes only marginally, from 15.93 to 15.86, and the total distance remains nearly unchanged, increasing slightly from 3434.86 to 3449.79. Similarly, the average transportation cost rises only from 4890.40 to 4912. These results indicate that higher variability does not significantly alter the spatial composition of the routes.

In contrast, the service-related measures deteriorate considerably as travel-time variability increases. The average service cost rises from 85.97 to 130.01 (an increase of 51.23%). This deterioration is also reflected in the timing indicators. Average expected earliness increases from 1189.00 to 1685.11 (an increase of 41.72%), while average expected lateness rises from 52.26 to 91.52 (an increase of 75.12%). The most pronounced effect is observed in overtime, which increases from 3.69 to 37.73 (an increase of 922.49%). These findings show that when the travel-time distribution becomes much more dispersed, the existing route plans become far less capable of preserving schedule quality, even though the transportation-related component of the solution remains relatively stable.

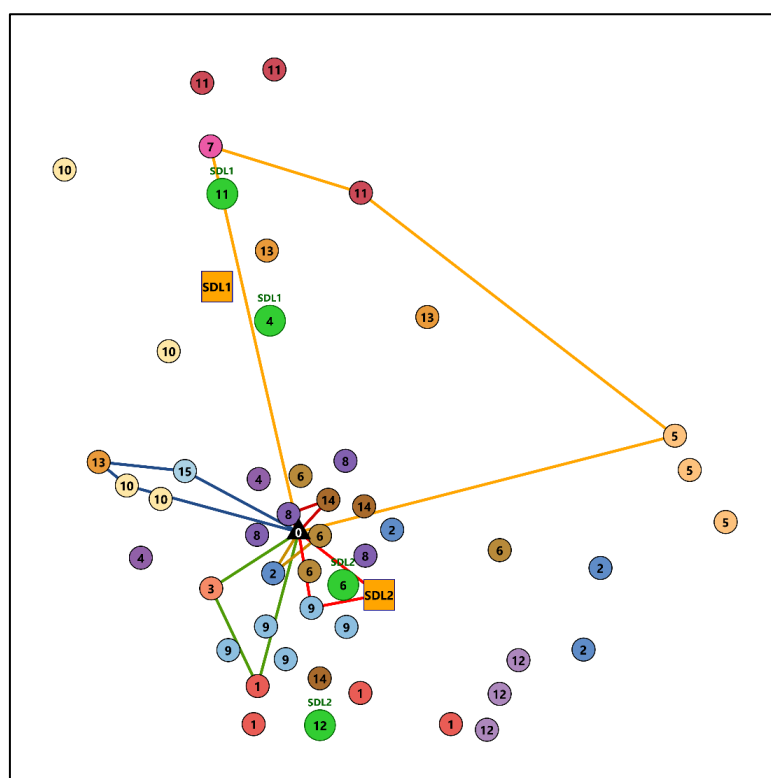
From a managerial perspective, these findings suggest that higher uncertainty primarily affects operational reliability rather than nominal routing efficiency. In other words, under highly variable travel conditions, the main risk is not a substantial increase in travel distance or fleet usage, but rather a severe degradation in schedule adherence and overtime performance. Therefore, in highly uncertain environments, logistics planners may need to incorporate additional buffer times, adopt more flexible delivery arrangements, or assign greater emphasis to service-related penalties within the objective function to better protect solution quality.

#### 5.4.3. Effects of the SDL Generation Policy

In practice, logistics companies may not want to define SDLs completely independently of their current operating plans. A more realistic approach is to learn candidate SDLs from the routes already observed in a no-SDL solution. This has a strong managerial appeal that the company first identifies route segments where several visits already occur in geographic proximity, then it turns these recurrent patterns into SDL opportunities. Such a policy can reduce implementation risk because SDLs are introduced where field operations already suggest natural consolidation points, rather than imposing entirely artificial meeting locations. Under this policy, the procedure starts from a feasible no-SDL solution represented as a set of vehicle routes, where each route is defined by a sequence of visited



exploit the routing structure observed in the baseline solution, and therefore the created SDLs may not be strongly aligned with the actual operational compatibility of customer visits. Figure 3 presents the SDL locations generated by the proposed route-based policy. Under this new policy, the SDLs are derived from the structure of the no-SDL solution, so that the shared locations are more consistent with the actual delivery tours. As a result, the total solution cost is further reduced to 1662.43, which corresponds to an improvement of 2.21% relative to the no-SDL case. This clearly shows that the proposed route-based SDL generation mechanism produces a better solution than both the no-SDL case and the initial SDL policy. In particular, compared with the initial SDL configuration in Figure 2 the proposed policy achieves an additional cost reduction of 22.78 units, highlighting the value of incorporating route information when constructing SDLs.

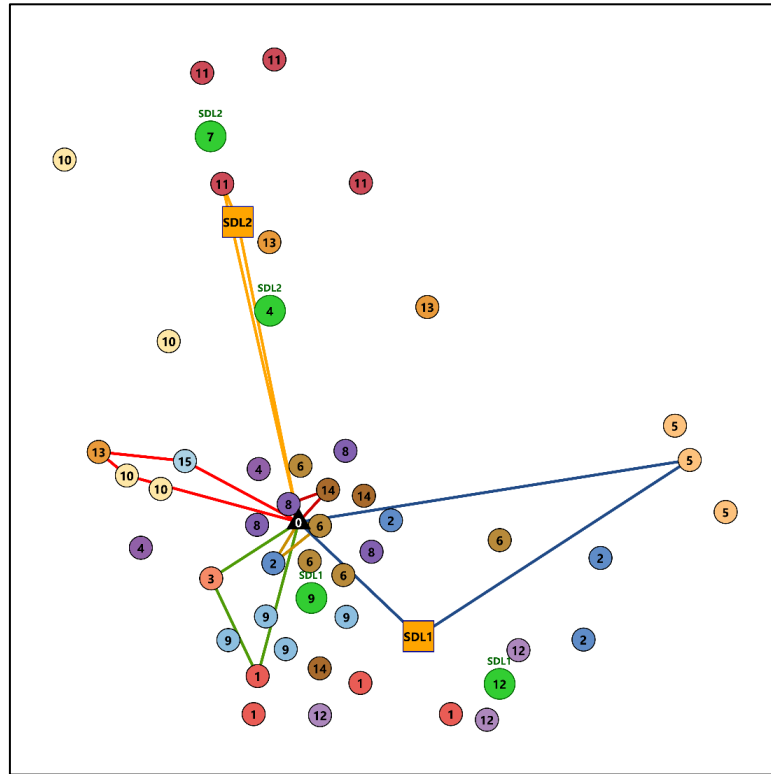


**Figure 2.** SDL locations generated by the existing policy for instance 1

Overall, the comparison among Figure 1 to Figure 3 shows that both SDL-based approaches improve upon the baseline no-SDL solution, but the proposed route-based SDL policy yields the best performance. This result indicates that SDL design should not be performed independently of the routing solution. Instead, exploiting the structure of the baseline delivery tours leads to more effective SDL placements and stronger cost improvements.

The results of 14 selected subset instances are provided in Table 13. The comparison between the two SDL policies shows that the proposed route-based construction rule is more effective than the existing SDL policy on average. Relative to the no-SDL baseline, the existing SDL policy reduces the average UB from 4976.37 to 4950.01, corresponding to an average improvement of 0.89%. The proposed new policy further decreases the average UB to 4858.79, yielding an average improvement of 2.96%.

Therefore, the new route-based policy provides a substantially larger average benefit than the existing SDL construction rule.



**Figure 3.** SDL locations generated by the proposed route-based policy for instance 1

**Table 13.** Comparison of UB values and improvement percentages under different SDL policies

Instance	No-SDL Case	Existing SDL		New policy	
	UB	UB	Improvement %	UB	Improvement %
1	1700.05	1685.21	0.87	1662.43	2.21
3	1537.25	1526.23	0.72	1427.59	7.13
7	2111.26	2028.56	3.92	2048.57	2.97
10	2393.62	2393.62	0.00	2316.22	3.23
15	3352.65	3265.27	2.61	3269.58	2.48
17	3186.95	3143.13	1.37	3123.79	1.98
21	6189.27	6189.27	0.00	6076.58	1.82
29	6239.82	6239.82	0.00	6171.62	1.09
36	13133.02	13133.02	0.00	12907.09	1.72
40	12048.51	12048.51	0.00	11960.22	0.73
42_v1	4527.26	4499.79	0.61	4387.64	3.08
49_v1	4709.75	4706.27	0.07	4477.54	4.93
44_v2	4212.43	4118.99	2.22	4028.03	4.38
48_v2	4327.36	4322.43	0.11	4166.15	3.73
Average	4976.37	4950.01	0.89	4858.79	2.96

A more detailed examination of the results also supports this conclusion. The existing SDL policy fails to improve some instances, including 10, 21, 29, 36, and 40, whereas the proposed route-based policy improves all instances. Its advantage is particularly visible in instances 3, 10, 42\_v1, 49\_v1, 44\_v2, and 48\_v2, where the improvement obtained by the proposed method is substantially greater than that of the existing SDL rule. For example, in instance 49\_v1, the existing policy achieves only a 0.07% improvement, while the proposed policy reaches 4.93%. Similarly, for instance 10, the existing policy provides no improvement, whereas the proposed method reduces the UB by 3.23%. These findings

indicate that SDLs become considerably more effective when their coordinates are derived from the route structure of the no-SDL solution rather than generated independently of it. From a managerial perspective, this result is important because it suggests that candidate SDLs need not be identified solely through static geometric considerations. Instead, companies can exploit existing delivery plans as operational evidence for defining more realistic and implementable consolidation points. In this way, the proposed policy transforms baseline routing solutions into a practical decision-support tool for identifying shared delivery opportunities while also achieving stronger cost reductions.

Overall, the three sensitivity analyses confirm that the proposed SVRPFD and SVRPFD-SDL framework reacts in a consistent and interpretable manner. The  $\mu$ -based analysis quantifies the trade-off between transportation efficiency and service quality, the CV-based analysis highlights the strong effect of uncertainty on timing reliability, and the SDL analysis shows that route-informed flexibility design can generate additional improvements beyond those obtained by a standard SDL construction policy.

## 6. Conclusions and Future Remarks

This paper introduced a Stochastic Vehicle Routing Problem with Flexible Deliveries, in which each customer can be served at one of several feasible delivery locations with associated time windows under uncertain travel times. The problem captures an important feature of modern last-mile logistics, where delivery flexibility and travel-time uncertainty must be considered jointly. To solve the problem, we proposed a solution approach based on column generation that integrates routing and delivery-location decisions while accounting for expected service penalties. Computational results showed that the proposed approach is able to produce high-quality solutions within reasonable runtimes and that delivery flexibility can lead to meaningful improvements in routing performance.

The sensitivity analyses further support the managerial relevance of the proposed framework. Overall, the three analyses show that the proposed models respond in a consistent and interpretable manner to changes in key problem settings. The  $\mu$ -based analysis quantifies the trade-off between transportation efficiency and service quality by showing how different weight settings shift the balance between routing cost and customer-oriented performance. The CV-based analysis highlights the pronounced impact of uncertainty on timing reliability, demonstrating that higher travel-time variability can substantially affect solution quality and routing behavior. Finally, the SDL analysis shows that designing shared delivery locations based on the no-SDL routing solution can provide additional benefits beyond those obtained under the standard SDL construction policy.

The study can be extended in several directions. Future research may consider richer uncertainty structures, such as time-dependent or correlated travel times, as well as additional operational features including electric vehicles and mobile parcel lockers. Another promising direction is the development of stronger exact and matheuristic solution approaches for larger-scale instances. Dynamic versions of the problem, where traffic conditions or customer delivery preferences evolve in real time, also deserve further investigation.

## Acknowledgments

This study was supported by Scientific and Technological Research Council of Türkiye (TÜBİTAK) under the Grant Number 223M102. The authors thank to TÜBİTAK for their supports.

## References

- Baldacci, R., Mingozzi, A., & Roberti, R. (2011). New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research*, 59(5), 1269-1283.
- Bomboi, F., Buchheim, C., & Prunte, J. (2022). On the stochastic vehicle routing problem with time windows, correlated travel times, and time dependency. *4OR*, 20(2), 217-239.
- Chapra, S. C., & Canale, R. P. (2011). *Numerical Methods for Engineers* (Vol. 1221). New York: Mcgraw-hill.
- DHL (2026) DHL delivery. Available at: <https://delivery.dhl.com/> (Accessed: 23 April 2026).
- Dragomir, A. G., Van Woensel, T., & Doerner, K. F. (2022). The pickup and delivery problem with alternative locations and overlapping time windows. *Computers & Operations Research*, 143, 105758.
- Duman, E. N., Taş, D., & Çatay, B. (2024). A bidirectional branch-and-price algorithm with Pulse procedure for the Electric Vehicle Routing Problem with flexible deliveries. *Transportation Research Part C: Emerging Technologies*, 165, 104699.
- Dumez, D., Lehuédé, F., & Péton, O. (2021). A large neighborhood search approach to the vehicle routing problem with delivery options. *Transportation Research Part B: Methodological*, 144, 103-132.
- Feillet, D., Dejax, P., Gendreau, M., & Gueguen, C. (2004). An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks: An International Journal*, 44(3), 216-229.
- Frey, C. M., Jungwirth, A., Frey, M., & Kolisch, R. (2023). The vehicle routing problem with time windows and flexible delivery locations. *European Journal of Operational Research*, 308(3), 1142-1159.
- He, Y., Qi, M., Zhou, F., & Su, J. (2020). An effective metaheuristic for the last mile delivery with roaming delivery locations and stochastic travel times. *Computers & Industrial Engineering*, 145, 106513.
- Janinhoff, L., Klein, R., & Scholz, D. (2024). Multitrip vehicle routing with delivery options: A data-driven application to the parcel industry. *OR Spectrum*, 46(2), 241-294.
- Jolfaei, A. A., & Alinaghian, M. (2024). Multi-depot vehicle routing problem with roaming delivery locations considering hard time windows: Solved by a hybrid ELS-LNS algorithm. *Expert Systems with Applications*, 255, 124608.
- Laporte, G., Louveaux, F., & Mercure, H. (1992). The vehicle routing problem with stochastic travel times. *Transportation Science*, 26(3), 161-170.
- Lombard, A., Tamayo-Giraldo, S., & Fontane, F. (2018). Vehicle routing problem with roaming delivery locations and stochastic travel times (VRPRDL-S). *Transportation Research Procedia*, 30, 167-177.
- Mancini, S., & Gansterer, M. (2021). Vehicle routing with private and shared delivery locations. *Computers & Operations Research*, 133, 105361.
- Ozbaygin, G., Karasan, O. E., Savelsbergh, M., & Yaman, H. (2017). A branch-and-price algorithm for the vehicle routing problem with roaming delivery locations. *Transportation Research Part B: Methodological*, 100, 115-137.

- Ozbaygin, G., & Savelsbergh, M. (2019). An iterative re-optimization framework for the dynamic vehicle routing problem with roaming delivery locations. *Transportation Research Part B: Methodological*, 128, 207-235.
- Pourmohammadreza, N., & Akbari Jokar, M. R. (2024). Efficient last-mile logistics with service options: A multi-criteria decision-making and optimization methodology. *International Journal of Industrial Engineering Computations*, 15(2), 367-386.
- Rajabi-Bahaabadi, M., Shariat-Mohaymany, A., Babaei, M., & Vigo, D. (2021). Reliable vehicle routing problem in stochastic networks with correlated travel times. *Operational Research*, 21(1), 299-330.
- Reyes, D., Savelsbergh, M., & Toriello, A. (2017). Vehicle routing with roaming delivery locations. *Transportation Research Part C: Emerging Technologies*, 80, 71-91.
- Sadati, M. E. H., Akbari, V., & Çatay, B. (2022). Electric vehicle routing problem with flexible deliveries. *International Journal of Production Research*, 60(13), 4268-4294.
- Oliveira, A. S., Kinable, J., Veelenturf, L., & Van Woensel, T. (2019). A scenario-based approach for the vehicle routing problem with roaming delivery locations under stochastic travel times. *Optimization Online*.
- Taş, D., Dellaert, N., Van Woensel, T., & De Kok, T. (2013). Vehicle routing problem with stochastic travel times including soft time windows and service costs. *Computers & Operations Research*, 40(1), 214-224.
- Taş, D., Gendreau, M., Dellaert, N., Van Woensel, T., & De Kok, A. G. (2014a). 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(3), 789-799.
- Taş, D., Dellaert, N., van Woensel, T., & De Kok, T. (2014b). The time-dependent vehicle routing problem with soft time windows and stochastic travel times. *Transportation Research Part C: Emerging Technologies*, 48, 66-83.
- Tilk, C., Olkis, K., & Irnich, S. (2021). The last-mile vehicle routing problem with delivery options. *OR Spectrum*, 43(4), 877-904.
- UPS (2026) UPS Access Point. Available at: <https://www.ups.com/us/en/business-solutions/expand-your-online-business/ups-access-point> (Accessed: 23 April 2026).
- Wang, S., Delage, E., & Coelho, L. C. (2024). Data-driven stochastic vehicle routing problems with dead-lines. *Les Cahiers du GERAD ISSN*, 711, 2440.
- Zhang, J., Lam, W. H., & Chen, B. Y. (2013). A stochastic vehicle routing problem with travel time uncertainty: trade-off between cost and customer service. *Networks and Spatial Economics*, 13(4), 471-496.
- Zhang, S. Z., & Lee, C. K. M. (2016, December). Flexible vehicle scheduling for urban last mile logistics: The emerging technology of shared reception box. In *2016 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)* (pp. 1913-1917). IEEE.