CIRRELT

CIRRELT-2025-38

The vehicle routing problem with driver scheduling

Razieh Mousavi
Jean-François Côté
Maryam Darvish

November 2025

Bureau de Montréal
Université de Montréal
C.P. 6128, succ. Centre-Ville
Montréal (Québec) H3C 3J7
Tél : 1-514-343-7575
Télécopie : 1-514-343-7121

Bureau de Québec
Université Laval,
2325, rue de la Terrasse
Pavillon Palasis-Prince, local 2415
Québec (Québec) G1V 0A6
Tél : 1-418-656-2073
Télécopie : 1-418-656-2624

The vehicle routing problem with driver scheduling

Razieh Mousavi, Jean-François Côté, Maryam Darvish

Department of Operations and Decision Systems, Université Laval, Québec, Canada and Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

Abstract: This paper addresses the vehicle routing problem with driver scheduling, which involves planning delivery routes while assigning work shifts to drivers whose individual availability must be respected. The problem extends the classical vehicle routing problem by incorporating driver availability, reflecting a growing operational concern in the logistics sector marked by persistent labor shortages and an increased need for flexible work arrangements. A mathematical model is developed to minimize total operational costs, including travel, driver shift, and outsourcing costs when some deliveries are assigned to third-party logistics providers. To handle large instances efficiently, a heuristic based on the Iterated Local Search algorithm is proposed. The algorithm integrates route optimization and driver scheduling decisions within a single framework. Computational experiments show that accounting for driver availability results in significant cost reductions and operational improvements. Even modest increases in driver availability significantly decrease the need for outsourcing, while full-day availability yields only marginal additional savings. These results offer practical managerial insights for logistics planners seeking to create a balance between operational efficiency and workforce constraints, as well as driver well-being.

Keywords: Vehicle Routing Problem, Driver scheduling, Iterated Local Search, Integrated routing, Driver availability.

Acknowledgements: This work was partly supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grants 2021-04037 and 2025-04195. We thank Compute Canada for providing high-performance parallel computing facilities.

---

* Corresponding author: Jean-Francois.Cote@fsa.ulaval.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2025

1 Introduction

In Canada, the pressures on last-mile logistics are mounting. The Canadian last-mile delivery market generated approximately USD 6.1 billion in 2023 and is projected to reach USD 8.9 billion by 2030 [Grand View Research, 2023]. This growth, driven largely by e-commerce expansion, has intensified operational pressures on carriers and delivery networks. However, the sector faces mounting labor constraints: Canada's trucking and logistics industry continues to experience chronic driver shortages [The Conference Board of Canada, 2021], and courier vacancies in some markets have surged by 22 % year over year [Lee, 2025]. To cope with these challenges, firms are rethinking how they organize work and deploy drivers.

Traditionally, logistics employees were assigned to fixed, non-flexible shifts; however, persistent labor shortages have encouraged companies to consider more adaptive scheduling policies to attract and retain employees. Many now rely on a hybrid workforce that combines full-time employees with contract drivers to increase delivery capacity. Partnerships with third-party platforms, such as Instacart and Grubhub, allow businesses to meet peak demand without maintaining large, permanent fleets. Meanwhile, others, including grocery chains and restaurants, have developed in-house delivery services to improve reliability and maintain control over service [Liu and Luo, 2023]. Some companies, including Amazon Flex and Deliv, guarantee a minimum pay per shift to ensure coverage during periods of fluctuating demand. Although this improves driver stability, it can also lead to idle capacity when volumes decline [Alnaggar et al., 2021].

Beyond the logistics sector, flexibility has become a defining feature of modern employment. Workers are increasingly seeking autonomy over their schedules to balance professional and personal responsibilities, while companies are using flexible or part-time arrangements to attract scarce talent. Yet such strategies complicate workforce planning: a large permanent workforce can absorb demand variations but entails high fixed costs,

whereas a smaller one risks overreliance on outsourcing [Mandal et al., 2025]. Managing this trade-off effectively requires embedding workforce considerations directly into operational decisions, such as routing and scheduling.

This paper addresses the integration of vehicle routing with driver shift scheduling problems. Routes are planned considering various constraints, such as time windows and vehicle capacity limits, as well as the availability of drivers. Each driver is assigned a shift that aligns with their availability. The main objective is to minimize the total cost, including shift and travel expenses. Additionally, unassigned demands are outsourced to a third-party logistics provider (3PL), incurring an additional fee. We refer to this problem as the Vehicle Routing Problem with Drivers Scheduling (VRPDS). To the best of our knowledge, this work is among the first to explicitly incorporate driver scheduling within a vehicle routing framework. We introduce a mixed-integer linear programming model (MILP) for the VRPDS and propose an Iterated Local Search (ILS) algorithm to solve large-scale instances. We also provide managerial insights on how changing attributes, such as drivers' availability, can affect the solution.

The remainder of this paper is organized as follows. In Section 2, we present an overview of the related literature and highlight our contributions. We provide the problem description and the mathematical formulation in Section 3. The proposed solution algorithm is elaborated in Section 4. Computational experiments and managerial insights are presented in Sections 5 and 6, and our conclusions follow in Section 7.

2 Literature review

To better position our study within the existing body of research, the following section reviews the literature related to the VRPDS, with a particular emphasis on how work *shifts* are defined and how *shift flexibility* is modeled. The review is structured in two

parts. The first focuses on staff scheduling, examining the literature on pure scheduling problems where routing is not explicitly considered. The second addresses integrated shift scheduling and routing, covering studies that jointly focus on shift design and vehicle routes. The section concludes by identifying the key conceptual and methodological gaps that motivate the formulation of the VRPDS.

In this context, a *personnel scheduling* or *rostering* problem refers to the process of constructing work timetables for employees so that an organization can satisfy the demand for its goods or services [Ernst et al., 2004]. The rostering process typically includes a shift-scheduling component, which determines which shifts are to be worked and how many employees should be assigned to each shift to meet operational requirements. Within such problems, the notion of *scheduling flexibility* plays a key role. Flexibility can manifest in various forms, such as varying shift lengths, flexible break placement (breaks starting within a time window rather than at a fixed time), or flexible start times (shifts beginning at different times during the day on a predefined grid) [Rekik et al., 2010]. Understanding how these elements are defined and modeled provides a foundation for examining the more complex integration of routing and scheduling decisions that characterizes the VRPDS.

2.1 Personnel scheduling

Given the extensive body of literature on personnel scheduling, our review will focus exclusively on the most pertinent studies that address flexibility within the scheduling aspect of the problem. Van den Bergh et al. [2013] provide a comprehensive review of personnel scheduling, highlighting how flexibility has been incorporated in several studies. They distinguish between fixed parameters, where start times and shift lengths are predetermined, and definable parameters, where these values are selected from an allowable set during the scheduling process. This distinction captures two dominant modeling approaches to shift flexibility in the literature. For instance, in the staff scheduling problem

at the United States Postal Service, as cited in Bard et al. [2003], both full-time and part-time shifts are defined. In the part-time case, twelve possible start times and five different shift lengths are considered. For full-time employees, nine possible start times are allowed with a fixed shift length. Similarly, Gutjahr and Rauner [2007] investigated a nurse scheduling problem in which nurses could propose several alternative start times and shift lengths that reflected their individual preferences. Other examples include Aickelin and Dowsland [2004] and He and Qu [2012], which treat shift structures as given and focus on assignment and constraint satisfaction rather than modeling flexibility.

A common observation from the above literature is that all models are based on an explicit representation of a shift. In contrast, some studies adopt implicit shift modeling, where shifts are not listed upfront but are generated by choosing start times and durations within specified bounds. Brunner et al. [2009] developed a flexible shift scheduling model for physicians based on a German university hospital case study. Unlike traditional approaches, their model employs implicit shift modeling, which allows for flexible start times and durations within defined constraints. The planning horizon spans multiple weeks, divided into one-hour periods. Key features include: shifts can start at designated periods and range from a minimum to a maximum length. Rekik et al. [2010] considered flexibility in terms of shift starting time and length, as well as the number, duration, and placement of breaks within each shift. They proposed two implicit models and solved them using a commercial solver.

Table 1 summarizes how shifts are modeled in personnel scheduling. Shift scheduling is used in various applications, including postal operations, nursing, physician services, and airport ground staff. A notable gap in the literature is the limited attention given to implicit shift modeling and staff-defined shifts, and, to the best of our knowledge, their combination has not been explored. Few models allow employees to play an active role in proposing or defining their shifts while the schedule itself is generated implicitly from bounded start or end-time windows.

Table 1: Summary of shift modeling structure in personnel scheduling literature

Study	Shift modeling approach	Shift features			Application
		Start time	Length	Shift definer	
Bard et al. [2003]	Explicit	Definable	Definable & fixed	Scheduler	Postal service
Aickelin and Dowsland [2004]	Explicit	Fixed	Fixed	Scheduler	Nurse scheduling
Gutjahr and Rauner [2007]	Explicit	Definable	Definable	Staff	Nurse scheduling
Brunner et al. [2009]	Implicit	Within bounds	Within bounds	Scheduler	Physicians scheduling
Rekik et al. [2010]	Implicit	Within bounds	Within bounds	Scheduler	Air-traffic control agency
He and Qu [2012]	Explicit	Fixed	Fixed	Scheduler	Nurse scheduling
Wang et al. [2023]	Implicit	Within bounds	Within bounds	Scheduler	Staff scheduling at airports
This paper	**Implicit**	**No bounds**	**Within bounds**	**Scheduler & staff**	**Last mile delivery**

2.2 Shift scheduling and routing

Shift scheduling in the routing literature can be categorized by how shifts are modeled. It may involve explicitly defined shifts, in which specific shift types are predetermined and assigned to personnel. Alternatively, it can rely on implicit shift modeling, in which the shift boundaries (start and/or end times) are treated as decision variables within the routing process. A combined approach is also possible, where start times are explicitly set while shift duration or end times remain implicitly determined. When both start and end times are fixed and there is no predefined set of shifts to select from, the problem is not considered shift scheduling. Furthermore, existing studies differ in focus: some explicitly minimize shift-related costs in the objective function, while others prioritize routing performance or service quality, treating shifts as secondary constraints.

Several studies use a hybrid representation with explicit start times and implicit durations. For instance, Ren et al. [2010] study a multi-period, multi-shift VRP with overtime in healthcare logistics, where each vehicle performs a single route per shift with fixed start times but flexible end times, which may extend through overtime when demand peaks near shift changes. Their objective function minimizes travel, wages, overtime, and outsourcing costs. Similarly, Frohner and Raidl [2021] address a dynamic, stochastic VRP

with delivery deadlines, in which drivers may run multiple closed routes per day. Here, shift start times are fixed, while shift end times remain flexible. The primary objective is to avoid tardiness, and the secondary objective is to reduce labor and travel costs. In both cases, the objective does not explicitly minimize shift costs. In contrast, De Bruecker et al. [2018] work with explicit predefined shifts in waste collection, distinguishing between cheaper peak-hour shifts and more expensive non-peak shifts. Their model assigns routes to these shift types to directly minimize weekly labor costs, making shift-related costs an explicit target within the objective function.

When staff availability is considered, it may be represented in several ways. In healthcare applications, availability is commonly modeled through time windows. For example, Grenouilleau et al. [2019] address home healthcare routing in which caregivers define their working days and daily availability windows. Route construction implicitly determines departure times within these windows, while the objective function minimizes routing costs, overtime, and idle time. In delivery applications, availability is modeled differently, often through occasional drivers (OD) or crowdsourced delivery (CD) couriers. These systems rely on a hybrid workforce that combines full-time employees with independent couriers who declare their availability to accept tasks [Yang et al., 2024]. This has led to rapid growth in crowdsourced delivery research (e.g., Mancini and Gansterer [2024], Zhang et al. [2025], Barbosa et al. [2023]), where the availability of such couriers becomes a key operational constraint. Alnaggar et al. [2021] provide a comprehensive review of routing and scheduling approaches in CD platforms and emphasize that handling driver availability remains a major challenge. In these systems, availability is uncertain and typically follows an accept-or-reject pattern. Ulmer and Savelsbergh [2020] introduce the workforce scheduling problem with unscheduled drivers, where shift schedules (start times and durations) are determined for in-house drivers operating alongside uncertain CDs, to minimize total working hours while satisfying service-level requirements. Behrendt et al. [2023] address a similar problem: scheduled couriers sign up for shifts before operations

and are centrally dispatched to multiple tasks during their shift, whereas ad-hoc CDs arrive during the day and choose one order at a time. Their model determines the number of scheduled couriers required in each period to minimize the total cost of scheduled couriers, ad-hoc CDs, and expired orders.

Table 2 summarizes the literature on shift scheduling and routing problems according to shift types, cost considerations, staff availability, and application context.

Table 2: Positioning VRPDS against related routing and scheduling studies

Study	Staff availability		Shift flexibility		Cost in objective		Shift	Application
	CD	Time window	Start	End	Shift	Rout	definer	domain
Frohner and Raidl [2021]	–	–	–	✓	–	✓	Scheduler	Online store
Ren et al. [2010]	–	–	–	✓	–	✓	Scheduler	Healthcare provider
De Bruecker et al. [2018]	–	–	–	–	✓	–	Scheduler	Waste collection
Grenouilleau et al. [2019]	–	✓	✓	✓	–	✓	Staff/scheduler	Healthcare
Ulmer and Savelsbergh [2020]	✓	–	✓	✓	–	–	Scheduler	Same day delivery
Behrendt et al. [2023]	✓	–	–	–	✓	–	Scheduler	Same day delivery
This paper	–	✓	✓	✓	✓	✓	**Staff/scheduler**	**Last mile delivery**

2.3 Literature gap and contributions

As summarized in Table 2, existing studies primarily differ in how they represent workforce control, shift flexibility, and the degree of integration between routing and scheduling decisions. In most formulations, driver availability is predetermined or simplified into an accept–reject decision, without modeling when a driver can actually start or finish work. Moreover, shifts are typically selected from a fixed list of time intervals rather than being generated from individual availability profiles. As a result, planned schedules often fail to align with actual workforce constraints and operational flexibility, limiting both realism and managerial applicability.

Considering these gaps, our contributions are as follows: 1. We introduce the Vehicle

Routing Problem with Driver Scheduling (VRPDS), a VRP variant in which each driver's shift is generated endogenously within declared availability windows, rather than selected from a predefined list of shifts. 2. We propose a joint optimization framework that simultaneously determines driver schedules and vehicle routes, explicitly linking shift timing, route feasibility, and total operational cost. 3. We develop a mixed-integer programming (MIP) formulation strengthened with valid inequalities to solve small instances exactly, and an ILS-based heuristic tailored to larger instances through neighborhoods that adjust routes and shift durations jointly. 4. We conduct computational experiments to assess solution quality and efficiency across different instance sizes, showing that the proposed approach scales effectively. 5. We provide managerial insights on the operational and economic value of flexible shift generation, demonstrating improvements in cost, driver utilization, and service coverage compared with fixed-shift systems.

3 Problem description and mathematical formulation

The VRPDS involves fulfilling the demand of a set of customers within a single day using a fleet of vehicles, each operated by a driver and based at the depot. The terms "driver" and "vehicle" are used interchangeably throughout this paper. The problem is defined on a graph $G = (V, A)$, with $V = \{0, 1, ..., n, n+1\}$, the set of nodes and arc set $A = \{(i, j) \mid i, j \in V,\ i \neq j\} \setminus \{(n+1, 0)\}$. The starting and ending depots are represented by 0 and $n+1$, respectively, and $N = \{1, ..., n\}$ is the set of customer nodes. Let $P \subseteq V$ be a subset of nodes; the in-arcs and out-arcs of P are defined as $\delta^-(P) = \{(i, j) \in A : i \notin P, j \in P\}$ and $\delta^+(P) = \{(i, j) \in A : i \in P, j \notin P\}$.

Each customer $i \in N$ is associated with a demand quantity q_i, a service time s_i, and a time window $[\alpha_i, \beta_i]$ within which the delivery must start. The depot also operates within its time window, denoted as $[\alpha_0, \beta_0]$ for node 0 and $[\alpha_{n+1}, \beta_{n+1}]$ for node $n+1$ with $\alpha_0 = \alpha_{n+1}$ and $\beta_0 = \beta_{n+1}$. Time window constraints are considered hard, meaning

that no deliveries can be made outside these specified periods. The travel cost between nodes i and j is denoted by c_{ij}, while the travel time is t_{ij}, and the euclidean distance is represented as d_{ij}. We assume $c_{ij} = t_{ij} = d_{ij}$ throughout the study. In cases where the company's fleet cannot fulfill all customers, a 3PL covers unmet demands, incurring an additional cost of $f \times D$ per delivery where f is a coefficient and $D = \max_{i,j \in V} d_{ij}$.

Each vehicle $k \in K$, with a capacity Q, starts at the depot and must return to it before time T ($T = \beta_{n+1}$). Each driver k is available during the interval $[a_k, b_k]$, with $0 \leq a_k < b_k \leq T$, and receives an hourly salary of h_k. Drivers need to be assigned to a shift, which is an uninterrupted time interval defined by its start and end times. A shift $s = [o_s, e_s]$ is defined by its start time o_s and end time e_s, and must respect drivers availability periods and company rules for shift duration stating that each driver can work for a maximum of $\bar{d}$ hours, where $\bar{d} < T$ and $b_k - a_k \leq \bar{d}$, and a minimum of $\bar{c}$ hours, where $\bar{c} \leq \bar{d}$. Let S_k be the set of feasible shifts for driver k where $S_k = \left\{[o_s, e_s] \mid a_k \leq o_s < e_s \leq b_k,\ \bar{c} \leq e_s - o_s \leq \bar{d}\right\}$. We introduce the parameter g as the time increment between successive potential shift start times, thereby aligning all start times at fixed intervals of length g.

We define the following variables to model the problem. Variable y_{ks} is binary and takes value 1 if driver k is assigned to shift s, and 0 otherwise. x_{ijk} is a binary variable that equals 1 if vehicle k uses arc $(i, j) \in A$, and 0 otherwise. u_j denotes the service start time at node j. Finally, w_i is a binary variable indicating whether the 3PL serves customer i.

The objective of the VRPDS is to minimize the total operational cost, which includes both delivery costs and driver salaries, while ensuring that all customers are visited. The problem is formulated as follows.

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ijk} + \sum_{i \in N} f D w_i + \sum_{k \in K} \sum_{s \in S_k} (e_s - o_s) h_k y_{ks}, \tag{1}$$

$$\text{s.t.} \sum_{s \in S_k} y_{ks} \leq 1 \qquad k \in K \tag{2}$$

$$\sum_{k\in K}\sum_{j\in\delta^+(i)} x_{ijk} = 1 - w_i \qquad i \in N \qquad (3)$$

$$\sum_{i\in\delta^-(j)} x_{ijk} - \sum_{i\in\delta^+(j)} x_{jik} = 0 \qquad k \in K, j \in N \qquad (4)$$

$$\sum_{j\in\delta^+(0)} x_{0,j,k} = \sum_{s\in S_k} y_{ks} \qquad k \in K \qquad (5)$$

$$\sum_{j\in\delta^-(n+1)} x_{j,n+1,k} = \sum_{s\in S_k} y_{ks} \qquad k \in K \qquad (6)$$

$$\sum_{i\in N}\sum_{j\in\delta^+(i)} q_i x_{ijk} \le Q \qquad k \in K \qquad (7)$$

$$\alpha_j \le u_j \le \beta_j \qquad j \in N \qquad (8)$$

$$u_j \ge \sum_{s\in S_k}(o_s + t_{0j})y_{ks} - T(1 - \sum_{(i,j)\in A} x_{ijk}) \qquad j \in N, k \in K \qquad (9)$$

$$u_j \le \sum_{s\in S_k}(e_s - t_{0j} - s_j)y_{ks} + T(1 - \sum_{(i,j)\in A} x_{ijk}) \qquad j \in N, k \in K \qquad (10)$$

$$u_j \ge u_i + (t_{ij} + s_i)\sum_{k\in K} x_{ijk} - T(1 - \sum_{k\in K} x_{ijk}) \qquad (i,j) \in A \qquad (11)$$

$$u_j \ge \sum_{k\in K}\sum_{i\in N\setminus\{0,j\}} (t_{0i} + t_{ij} + s_i + a_k)\, x_{ijk} \qquad j \in N \qquad (12)$$

$$x_{ijk} \in \{0,1\} \qquad k \in K, (i,j) \in A \qquad (13)$$

$$w_i \in \{0,1\} \qquad i \in N \qquad (14)$$

$$y_{ks} \in \{0,1\} \qquad s \in S_k, k \in K \qquad (15)$$

$$u_j \in \mathbb{R}_{\ge 0} \qquad j \in V. \qquad (16)$$

The objective function (1) minimizes the total costs, including shift, routing, and 3PL delivery costs. Constraints (2) impose that a driver must be assigned to at most one shift. Constraints (3) guarantee that each customer is assigned to exactly one route if a driver serves it; otherwise, it will be assigned to the 3PL. Constraints (4) are flow conservation constraints at every intermediate node. Constraints (5)–(6) ensure that, for each driver assigned to a shift, there is a valid path by enforcing departure from and return to the

depot. Constraints (7) ensure that the vehicle capacity is respected. Constraints (8)-(11) define the timing of the visits and time window at the nodes. Constraints (12) are valid inequalities to force the visit time at any customer to be no earlier than the driver's shift start plus the travel and service time along any chosen arc. Finally, constraints (13)–(16) define the nature and domain of the variables.

4 Solution algorithm

The VRPDS is NP-hard, as it generalizes the classical VRPTW by introducing endogenous driver scheduling. Exact methods are thus impractical for larger instances. To address this challenge, we develop a tailored heuristic that combines an Iterated Local Search (ILS) framework with an embedded assignment optimization for driver–route pairing. While ILS has shown strong performance on large-scale routing problems [Subramanian et al., 2012, Mancini et al., 2021], our contribution lies in adapting and extending it to simultaneously handle routing, scheduling, and shift-cost decisions.

A solution to the VRPDS is represented as a set of routes $r_k = (i_1^k, i_2^k, \ldots, i_{l_k}^k)$, where each route is assigned to a driver k with a feasible shift $s_k = [o_k, e_k]$ that respects the driver's availability and company work-hour rules. Unserved customers are outsourced to the 3PL operator.

The algorithm begins by generating an initial feasible solution. Then, an initial local search procedure is applied to improve the solution. The ILS framework then perturbs the current solution to explore new neighborhoods of the solution space. After the perturbation, a local search is applied to produce a further improved solution. Finally, an assignment problem is solved to minimize shift costs, which reassigns routes to drivers. The perturbation, local search, and reassigning routes to drivers are repeated iteratively until a termination condition is met.

Our ILS is tailored to the integrated nature of VRPDS in three ways. First, every local move is evaluated based on the total cost, which comprises 3PL, travel, and shift-related expenses. This ensures that route changes are considered alongside staffing and outsourcing implications, rather than in separate phases. Second, we use forward time slack after each move or perturbation to identify the smallest feasible shift for a driver that accommodates the constructed route. This approach will precisely adjust the shift's start and end times and ensure all constraints are satisfied, including driver availability. Third, the pairing between routes and drivers is reoptimized at each iteration by solving an assignment model that determines the most cost-effective driver for each route. This procedure ensures that the overall shift cost across all routes is minimized while maintaining feasibility within the scheduling framework.

In the following sections, we describe the process for determining the optimal shift for each driver and route, and provide a detailed explanation of the proposed ILS framework.

4.1 Shift optimization strategy

This section outlines the strategy for determining the optimal shift for a driver based on their assigned route. The best shift has the shortest duration among all feasible shifts. To identify it, we adapt the concept of forward time slack, a principle widely employed in the vehicle routing literature (see Savelsbergh [1992] and Cordeau et al. [2004]).

For simplicity, we reindex the nodes visited on route r_k as $(0, 1, \ldots, l_k)$, where 0 and l_k represent the depot. The route is assumed to start as early as possible. At each node i on this route, we calculate the arrival time A_i, the waiting time $W_i = \max\{0, \alpha_i - A_i\}$, and the beginning of service time B_i. The forward time slack F_i^k measures how much the start of service at node i can be delayed without violating the time windows of the route. Savelsbergh [1992] computes it as $F_i^k = \min_{i \leq j \leq l_k} \left\{ \beta_j - \left(B_i + \sum_{i \leq p < j} t_{p,p+1} \right) \right\}$.

In our context, calculating forward time slack requires an adjustment to incorporate

service times. This ensures that both travel and service times are properly considered when determining the slack for a given route. The modified time slack can then be calculated as $F_i^k = \min_{i \leq j \leq l_k} \left\{ \beta_j - \left(B_i + \sum_{i \leq p < j} t_{p,p+1} + \sum_{i \leq p < j} s_p \right) \right\}$.

To find the best shift, we must determine the impact of postponing the departure from the depot. By delaying the departure by the minimum forward time slack at the depot and the total waiting time along the route, we ensure that the final arrival time A_{l_k} at the route's endpoint remains unchanged. Specifically, we calculate the delay as: $\min \left\{ F_0, \sum_{0 < p < l_k} W_p \right\}$. Any delay beyond this value would result in a later arrival at the endpoint. Thus, the shortest route duration that maintains feasibility and avoids constraint violations is given by $A_{l_k} - \left(a_k + \min \left\{ F_0, \sum_{0 < p < l_k} W_p \right\} \right)$.

Knowing that a route is feasible if the driver's shift aligns with their full availability period, we aim to minimize the shift duration to reduce associated costs. To achieve this, we try to delay the route's start as much as possible while respecting all constraints. The forward time slack at the depot determines the latest possible start time, F_0, plus the driver's availability start time, a_k. This ensures the route remains feasible while minimizing the shift cost by shortening the duration. The largest feasible start time, which we call $Start$, is $\left\lfloor \frac{a_k + F_0}{g} \right\rfloor \times g$. The start time is rounded down to the nearest multiple of g minutes.

Assuming $W_T^k = \sum_{0 < p < l_k} W_p$, $Start$ can either be greater than $a_k + W_T^k$ or less than $a_k + W_T^k$. We can determine how it affects the end of the shift depending on which interval the $Start$ time falls into. If $a_k + W_T^k \leq Start$, then the end of the route will be delayed by $\Delta = Start - (a_k + W_T^k)$. Otherwise, the end time, End, will remain unchanged. Consequently, the End of the route is defined as follows:

$$End = \begin{cases} \left\lceil \frac{A_{l_k}}{g} \right\rceil \times g & \text{if } Start \leq a_k + W_T^k, \\ \left\lceil \frac{A_{l_k} + \Delta}{g} \right\rceil \times g & \text{if } Start \geq a_k + W_T^k. \end{cases}$$

However, End must be aligned to a standard schedule, meaning it should be divisible by g minutes. Therefore, the end of the route is achieved by $A_{l_k}^{new} = A_{l_k} + \max(\Delta, 0)$.

4.2 Initial solution

We use a constructive heuristic based on the cheapest feasible insertion to create the initial solution. To begin with, all customers are assigned to the 3PL, and drivers are given the largest possible shifts based on their availability. The list of candidate customers for insertion (LC) is initialized with all customers. The process begins by generating routes for drivers, during which customers are inserted from the LC into the drivers' routes.

In the cheapest insertion procedure, the change in total cost when inserting a node is calculated based on travel costs, shift adjustments, and 3PL costs, calculated as $C(j,k) = (c_{ij} + c_{jl} - c_{il}) - f + h_k\left[(e_k - o_k) - (e'_k - o'_k)\right]$ where $C(j,k)$ represents the cost of inserting an unassigned customer j into the route of driver k. The driver's optimal shift before inserting node j is denoted by $[o_k, e_k]$ and after inserting node j between nodes i and l, the updated shift is $[o'_k, e'_k]$, with o'_k and e'_k as the new start and end times, respectively. The algorithm evaluates each potential insertion by calculating $C(j,k)$ for every possible customer-to-route insertion, considering capacity constraints, time windows, and driver availability. The customer is inserted into the route offering the lowest feasible cost. If infeasible or cost-ineffective, the customer is reassigned to the 3PL.

4.3 Local Search

This section presents the local search procedures used to improve the routing solutions by iteratively refining the current routes through neighborhood exploration. We perform one intra-route and four inter-route local search procedures, which are improved by reassigning routes to drivers and solving an assignment problem at each iteration. The best improvement approach iteratively evaluates all possible moves within a defined neighborhood, selects the one that offers the biggest improvement to the current solution, and continues this process until no further improvements can be made. Evaluating a neighborhood solution involves finding the optimal schedule for each route and ensuring that

the schedule maintains route feasibility while minimizing the shift duration for the route.

4.3.1 Intra-route neighborhood structure

In intra-route local search, each route is improved by relocating its nodes. This process involves assessing the impact of moving nodes within the route to reduce shift and routing costs to potentially minimize or eliminate waiting times. The algorithm aims to find the best move within each route by evaluating all nodes and their possible relocations. For every non-empty route, it examines each node and assesses potential moves to determine which relocation minimizes the total cost. It calculates the cost change for each move and verifies its feasibility. If a feasible move reduces costs, it is selected as the best move for that route. After applying the best move, the algorithm updates the route and moves on to the next one. This process continues until no further cost-reducing moves can be found across any route.

4.3.2 Inter-route neighborhood structures

We propose four inter-route neighborhood structures. The solution space is exhaustively explored, i.e., all possible combinations are examined to find the best improvement.

Relocate Node: This local search evaluates relocating customers within existing routes, assigning them to available drivers, or shifting a node from an existing route to 3PL.

*2-opt**: Potvin and Rousseau [1995] introduced the 2-opt* for problems with time windows. We iterate over two routes at a time and, for each, exchange the right-hand portion of their routes starting from a selected customer.

Or-opt Move: The classical Or-opt method removes a sequence of two to four nodes and reinserts it at a new position [Groër et al., 2010]. However, we evaluate all chain length combinations and also relocate chains that include sequences of unassigned nodes.

Cross-exchange: It swaps two chains of customers between two routes [Taillard et al., 1997]. For each selection of two routes, chains ranging from 1 to L customers are considered for exchange, removing four edges and replacing them with four new ones.

The inter-route neighborhood structures are applied in the following order: Relocate Node, Or-opt, 2-opt*, and Cross-exchange.

4.4 Perturbation

The main role of the perturbation process is to modify a local optimal solution by transitioning from a current solution to a neighboring one. The perturbation process involves removing a certain number of nodes from a given solution and re-inserting them into other positions in the routes or assigning them to the 3PL. If the number of relocated nodes is too large, ILS behaves like random restarts; if it is too small, the search tends to return to the same recently visited local optimum [Lourenço et al., 2003].

We apply the concentric removal method used by Maximo et al. [2024]. First, we remove nodes and assign them to 3PL by randomly selecting a node and removing a set of δ nodes clustered around it, ensuring they remain closely grouped. Then, we check all the unassigned nodes and determine the best insertion for each. If not feasible due to time windows, capacity, or shift limitations, the affected nodes will be assigned to the 3PL. During the reinsertion process, the total cost is computed, including the shift cost.

4.5 Reassigning routes to drivers

After improving the solution through the local searches, we obtain a set of routes and assigned drivers. However, some available drivers may not have been assigned to any shifts. Since drivers have varying salaries, we can reduce shift costs by reassigning routes to the most cost-effective drivers, whether they are currently unassigned or already have

a route. This reassignment allows for the selection of the optimal shift for each generated route. To achieve this, we solve an assignment problem as follows:

Given a set of routes, $R = \{r_1, r_2, \ldots, r_m\}$ and the set of drivers, K, where some drivers may not currently be assigned to any shifts, we solve the problem (17)- (20). The objective is to minimize the cost of assigning routes to drivers and finding the best shifts. The term c_{rk} represents the cost associated with assigning route r to driver k. We calculate the shift cost for each route and assign it to different drivers, considering feasibility constraints such as availability and shift duration. If a shift is infeasible, a high penalty is assigned to it. The binary variable x_{rk} indicates the assignment status, where $x_{rk} = 1$ if route r is assigned to driver d, and $x_{rk} = 0$ otherwise.

$$\min \sum_{r \in R} \sum_{k \in K} c_{rk} x_{rk} \tag{17}$$

$$\text{s.t.} \sum_{k \in K} x_{rk} = 1 \qquad r \in R \tag{18}$$

$$\sum_{r \in R} x_{rk} \leq 1 \qquad k \in K \tag{19}$$

$$x_{rk} \in \{0, 1\} \quad r \in R, k \in K \tag{20}$$

Constraints (18) ensure that each route is assigned to exactly one driver, and constraints (19) guarantee that each driver is assigned to at most one route. Constraints (20) define the assignment's binary decision variables.

The optimal assignment is determined using the Hungarian algorithm [Kuhn, 2004] which checks if this solution reduces shift costs compared to existing assignments. Subsequently, the driver shifts and routes are updated based on the new optimal assignment.

5 Computational results

In this section, we present and analyze the results of the computational experiments to assess the performance of the proposed solution algorithm for the VRPDS. First, we introduce the instances in Section 5.1. Parameters and the tuning procedure are presented in Section 5.2. The performance of the proposed algorithm is compared against that of the MILP solver, and the results are presented in Section 5.3. The mathematical formulation and the proposed solution method are implemented in C++ using IBM ILOG CPLEX Concert Technology 22.11 as the MILP solver. Computational experiments are performed on an AMD EPYC 7532 CPU with a 2.4 GHz and a memory limit of 32 GB of RAM. A single thread and a one-hour time limit are set.

5.1 Instance generation

The literature has no benchmark instances for the VRPDS, therefore we have generated the following dataset for our experiments. An instance is identified as "n- $|K|$- Q" where n is the number of customers, $|K|$ the number of drivers, and Q the capacity of each vehicle. As shown in Table 3, instances are divided into two classes based on the number of customers: small and large. In the small class, instances with $n \in \{10, 20\}$ are served by $|K| = 5$ vehicles (capacity $Q = 200$), whereas those with $n \in \{30, 40, 50\}$ use $|K| = 10$ vehicles (capacity $Q = 250$). In the large class, instances with $n \in \{100, 200, 300, 400\}$ are handled by $|K| \in \{20, 30, 40, 50\}$ vehicles with capacities $Q \in \{300, 350, 400, 450\}$. Each instance is generated using four time windows: one hour, two hours, three hours, and a randomly selected time window among one, two, and three hours. Five instances are created for each set. Therefore, we run 180 instances in total.

Customers are randomly placed within a 100×100 square on a two-dimensional Euclidean plane, with the depot centrally located at coordinates $(50, 50)$. Vehicles travel at a speed

Table 3: Instance configurations

Instance class	n	$\|K\|$	Q
Small	10, 20	5	200
	30, 40, 50	10	250
Large	100	20	300
	200	30	350
	300	40	400
	400	50	450

of one unit of distance per minute. The travel time matrix is symmetric and respects the triangular inequality. The planning horizon is from 6 AM to 6 PM, or 0 to 720 minutes, with the depot's time window also covering these 12 hours. Service times for each customer are randomly assigned between 5 and 10 minutes, and demands range between 10 and 20 units. Each customer has a time window, with the start time randomly selected from the interval $(\alpha_0 + t_{0i}, \beta_{n+1} - t_{i(n+1)} - s_i)$, where α_0 and β_0 are the start and end times of the depot's time window, t_{0i} and $t_{i(n+1)}$ are the travel times between the depot and the customer, and s_i is the service time at the customer.

Drivers have fixed 4-hour availability periods, covering five time slots: [0- 240], [120- 360], [240- 480], [360- 600], and [480- 720]. These availabilities ensure full coverage of the time horizon. If there are 5 drivers, each is assigned a unique availability. If there are 10 drivers, each availability is assigned to two drivers, and this pattern continues as the number of drivers increases.

The salary for each driver is randomly generated between \$15 and \$20 per hour. For comparison, the U.S. Bureau of Labor Statistics reports a median wage of \$17.03 per hour (25th–75th percentile: \$12.45–\$22.42 per hour) for delivery-truck drivers as of May 2023 [U.S. Bureau of Labor Statistics, 2024]. Additionally, the minimum working hours per driver is 2 hours, and the maximum is 8 hours. Finally, the granularity of the shift

start intervals, denoted by g, is set to 30 minutes, allowing shifts to start every half hour.

All instances generated for this study and a dataset with the detailed results of our experiments can be found at https://sites.google.com/view/jfcote/.

5.2 Parameters tuning

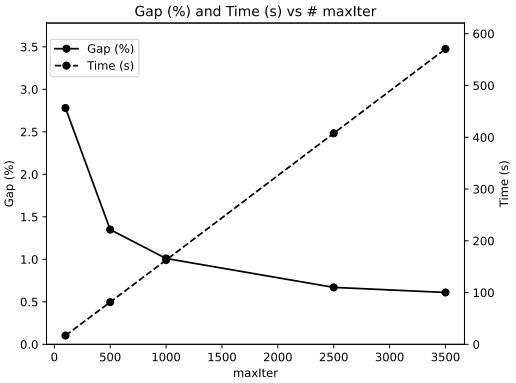
This section presents the parameter tuning process for the ILS used to solve the VRPDS. We perform parameter tuning on a training set of 20 instances, comprising ten small-sized and ten large-sized instances, with varying numbers of customers.

The primary parameter selected for tuning is the number of iterations before termination, while keeping the perturbation rate at 10% . The heuristic was executed with 100, 500, 1000, 2500, and 3500 iterations. For each iteration count, a single run was conducted.

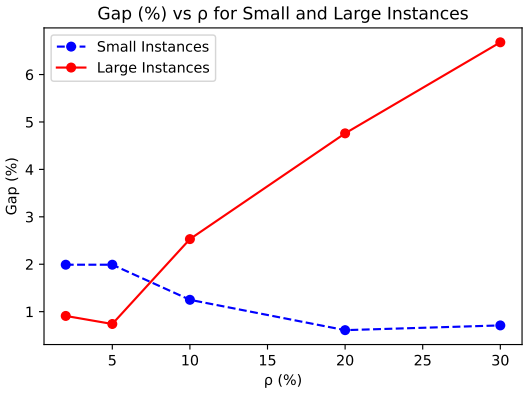
Figure 1a illustrates the average runtime and the average gap relative to the Best Known Solution (BKS) for each configuration. The BKS represents the best solution found across all iterations. The results indicate a significant reduction in the gap between the solution and the BKS when the number of iterations increases from 100 to 2500, followed by a more gradual improvement between 2500 and 3500 iterations. However, the execution time almost doubles with each increase in the number of iterations. Considering this trade-off, we selected 2500 iterations as the number of iterations ($maxIter$), as it makes a balance between solution quality and computational efficiency.

Next, in Figure 1b, we evaluate the effect of the perturbation rate ρ on both small and large instances, as our preliminary analysis revealed varying impacts across these instance sizes. We tested perturbation rates of 2%, 5%, 10%, 20%, and 30%. The results, presented in Figure 1b, show that for small instances, a perturbation rate of 20% achieves the smallest gap, whereas for large instances, a 5% rate yields the best performance. Based on these observations, we selected 20% for small instances and 5% for large instances as the optimal

perturbation rates.



(a) Effect of $maxIter$ on the performance of the ILS, $\rho = 10\%$.



(b) Effect of perturbation rate on the performance of the ILS for small and large instances.

Figure 1: Parameter tuning for $maxIter$ and the perturbation rate, ρ.

At each perturbation step, we modify $\delta = \rho \cdot n$ number of customers. The search is allowed to run for at most $maxIter = 2500$ iterations.

In the next phase of tuning, we focus on selecting the operators used for ILS. As introduced in Section 4.3, five local search operators are used. To assess their contribution, we conducted experiments on the 20 instances mentioned earlier, where the ILS was run without one operator at a time. Each experiment was run ten times, and the average gap relative to the BKS across all runs is presented in Figure 2 using boxplots. The results indicate that the *Relocate Node* operator is the most critical, as its removal leads to the highest average gap of approximately 10%. In contrast, the *2-opt** operator has the least influence on solution quality, with a gap of around 2.0%. The *Intra-route* operator ranks second in importance, followed closely by the *Cross-exchange* local search. The remaining operators contribute almost equally to the overall performance. Importantly, the smallest average gap of approximately 1.5% is achieved when all operators are employed. The two figures in Figure 2, provide complementary insights: Figure 2b, shows a better comparison

of operators by excluding the effect of *Relocate* operator. Preliminary experiments also showed that this sequence of operators is efficient.

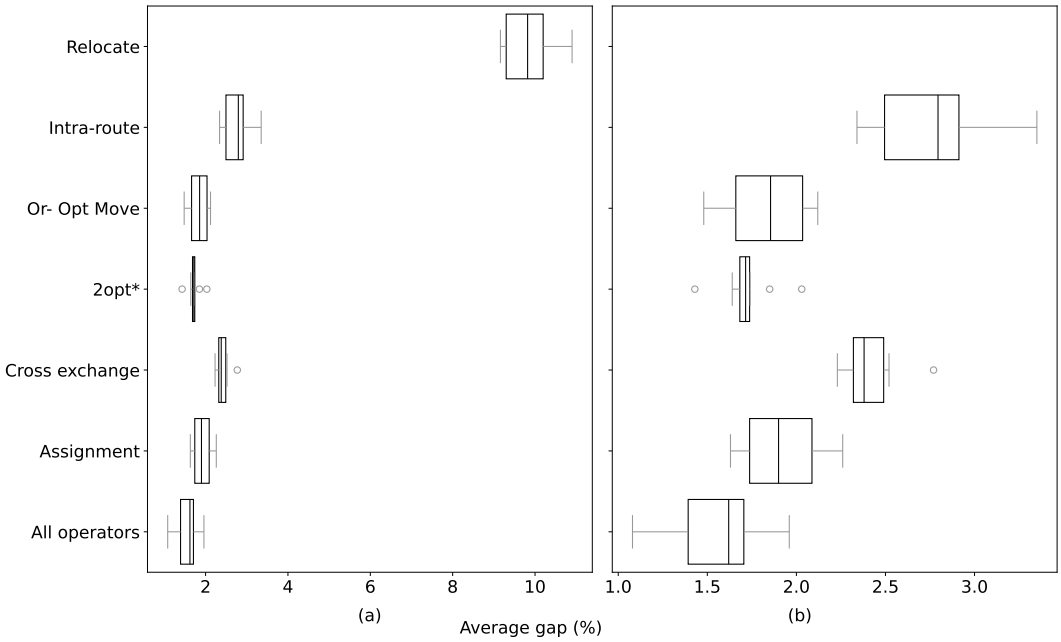


Figure 2: Contribution of each local search to the solution quality.

5.3 Results analysis

The computational results are summarized in Tables 4 and 5. A time limit of 3600 seconds was imposed for both methods, with the ILS also employing a stopping criterion of 2500 iterations. In these tables, "TW" refers to the length of time windows, which can be one, two, three hours, or a random duration. "UB" and "LB" represent the average upper and lower bounds obtained by CPLEX. The column labeled "Opt" reports the number of instances solved to optimality, while "T(s)" indicates the average computational time in seconds. We observe that while ILS consistently provides high-quality solutions within the time limit, CPLEX can solve 53 out of 100 small instances to optimality. For instances with less than 30 customers, all five are solved to optimality, but for 50 customers, only

one out of five.

Table 4: Results from CPLEX

	TW=1				TW=2				TW=3				TW = Random			
Instances	UB	LB	Opt	T(s)	UB	LB	Opt	T(s)	UB	LB	Opt	T(s)	UB	LB	Opt	T(s)
10-5-200	689.4	689.4	5	0.1	605.1	605.1	5	0.2	566.6	566.6	5	1.8	660.5	660.5	5	0.1
20-5-200	1271.8	1271.8	5	0.4	1027.4	1027.4	5	14.9	929.5	929.5	5	656.1	1065.5	1065.5	5	21.0
30-10-250	1687.0	1687.0	5	63.4	1331.6	1207.6	1	3,453.0	1155.5	831.2	0	3,576.5	1356.1	1242.4	1	3,496.8
40-10-250	2048.9	2048.9	5	682.8	1645.0	1278.8	0	3,575.0	1536.3	955.3	0	3,570.0	1686.8	1320.5	0	3,573.1
50-10-250	2574.7	2459.7	1	3,043.8	1945.5	1346.3	0	3,573.9	1884.3	964.8	0	3,575.4	2165.8	1499.3	0	3,573.2
Average/ Total	1654.4	1631.4	21	758.1	1310.9	1093.0	11	2,123.4	1214.4	849.5	10	2,276.0	1386.9	1157.6	11	2,132.8

As shown in Table 4, under the 1-hour time window, 21 out of 25 instances were solved to optimality, with the four unsolved cases all belonging to the *50-10-250* set. For both the 2-hour and random time windows, 11 out of 25 instances reached optimality. Under the 3-hour time window, only 10 instances were solved. Overall, computational time increases with both the length of the time windows and the number of customers.

To compare the solutions obtained by the proposed algorithm with those from CPLEX, we define the *Gap* as the percentage difference between the UB and $\bar{Z}$, which is the average cost obtained by the algorithm over five runs. The gap is calculated as $Gap = 100 \times (\bar{Z} - UB)/UB$. Table 5 presents the gap and execution time for each instance, considering the four time window scenarios for instances with up to 100 customers, as CPLEX could not provide an upper bound for larger instances.

Table 5: Summary of results for the ILS

	TW=1		TW=2		TW=3		TW = Random	
Instances	Gap(%)	T(s)	Gap(%)	T(s)	Gap(%)	T(s)	Gap(%)	T(s)
10-5-200	2.10	1.63	2.03	1.67	4.28	1.63	0.97	1.66
20-5-200	3.84	2.50	5.30	2.58	6.49	2.74	5.14	2.71
30-10-250	4.27	4.80	4.07	5.04	3.57	5.37	4.93	4.88
40-10-250	5.56	6.67	5.53	7.42	**-1.22**	7.81	1.65	7.47
50-10-250	9.21	8.81	4.90	10.33	**-10.89**	11.94	**-1.20**	10.11
100-20-300	6.28	30.16	**-22.29**	35.44	**-64.14**	38.95	**-36.12**	34.52
Average	5.21	9.10	-0.08	10.41	-10.32	11.41	-4.10	10.22

For small instances (e.g., *10-5-200* and *20-5-200*), ILS produces solutions that are close to those obtained by CPLEX, with small gaps (e.g., 2.1% and 3.84% for $TW = 1$). These differences remain marginal across different time window settings, indicating that while CPLEX finds slightly better solutions, ILS remains competitive with significantly shorter computation times. As the problem size increases (e.g., *50-10-250* and *100-20-300*), the heuristic's performance varies depending on the time windows. For instance, at $TW = 3$, it achieves a large improvement of -64.14% for instance 100-20-300. On average, the heuristic achieves competitive results, with an overall gap of 5.21% at $TW = 1$, improving to -0.08% at $TW = 2$ and reaching -10.32% at $TW = 3$, where it significantly outperforms CPLEX. The execution time remains relatively low across all time window, confirming the efficiency of the ILS, particularly for larger problem instances where computational speed is a key factor.

For larger instances, since the solver did not provide an upper bound, we analyzed the average ($\bar{Z}$), best, and worst solutions, and stability gap over five runs to assess solution quality. Table 6 shows a summary of the results for large-sized instances across different time windows. The stability gap represents the percentage difference between the average

Table 6: Analysis of results for large-sized instances

# Customers	200				300				400			
TW	$\bar{Z}$	Best	Worst	Stability Gap	$\bar{Z}$	Best	Worst	Stability Gap	$\bar{Z}$	Best	Worst	Stability Gap
1h	6739.9	6408.8	7065.9	-5.18%	9207.8	8881.4	9473.8	-3.67%	11127.4	10835.1	11472.7	-2.70%
2h	4926.6	4787.9	5082.8	-2.91%	6340.0	6236.8	6454.6	-1.66%	7615.6	7493.4	7722.0	-1.65%
3h	4135.1	4051.2	4237.2	-2.08%	5449.6	5306.6	5596.7	-2.70%	6669.8	6534.7	6824.2	-2.07%
Random	5083.8	4960.4	5227.2	-2.49%	6681.2	6524.2	6844.2	-2.41%	8118.9	7989.5	8252.9	-1.61%
Total average	5221.4	5052.1	5403.3	-3.17%	6919.6	6737.3	7092.3	-2.61%	8382.9	8213.2	8568.0	-2.01%

and best solutions. It decreases slightly as the number of customers increases, from 3.2% for 200 customers to 2.6% for 300 customers and 2.0% for 400 customers. Considering the impact of customer time windows, instances with shorter TW (1h) have the highest variability, with an average gap of 5.2% for 200 customers, compared to 3.7% for 300 customers and 2.7% for 400 customers. As TW increases to 2h and 3h, the gap narrows, indicating that larger time windows improve solution stability. Overall, larger instances and longer time windows yield more stable solutions across multiple runs.

6 Managerial insights

The primary goal of this section is to provide managerial insights on how changes in policies regarding shift duration, 3PL cost, and drivers' availability can impact costs and the solution. These insights can help managers make informed decisions to enhance their supply chain's performance. In what follows, we test 50 instances with 10, 20, 30, and 40 customers with a known optimal solution.

6.1 Impact of driver availability on shift scheduling and costs

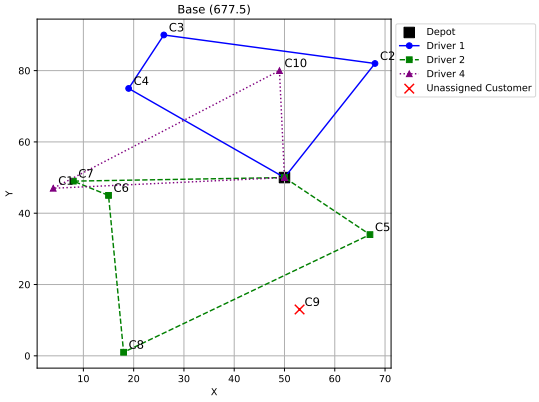
A dilemma for managers in last-mile operations is striking a balance between operational efficiency and workforce flexibility. Our results indicate that increasing driver availability

Table 7: Cost analysis for different availability periods

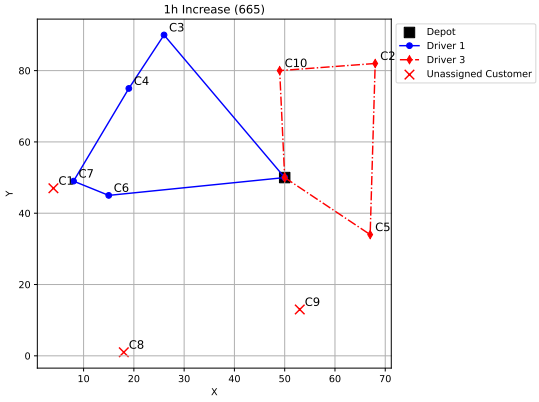
		Increase				
Cost	Availability: 4h	1h	2h	3h	4h	Full availability
Shift	249.0	258.2	271.8	277.4	284.0	286.6
Travel	625.5	613.4	615.4	602.9	599.1	593.2
3pl	1111.2	147.8	87.04	74.4	63.1	52.9
Total	1985.8	1019.5	974.2	954.8	946.2	932.9

can significantly reduce overall costs, primarily by reducing reliance on 3PL services. For example, as shown in Table 7, when availability increases from 4 hours to full-day, total costs drop by over 50%. However, the marginal cost savings decline sharply after a 2-hour extension, with only a 2.3% gain from moving to full-day availability. For decision-makers, this suggests a sweet spot of around 6 hours of availability, as it delivers most of the cost benefits while avoiding the need for lengthy or inflexible shifts that may reduce driver satisfaction or breach labor constraints. Beyond this point, the gains become marginal, and the operational burden (e.g., managing longer shifts, risk of fatigue, or employee dissatisfaction) may outweigh the savings. These results suggest that moderate extensions in shift availability, possibly through voluntary overtime or flexible scheduling policies, could deliver considerable savings with minimal negative impact. However, pushing for full availability may not be justified unless cost reduction is the overriding priority and labor conditions allow it.

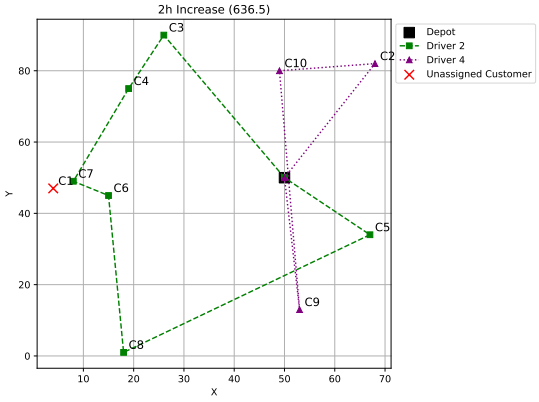
An example of the effect of availability is illustrated in Figure 3, which shows three solutions after increasing availability by 1 hour and 2 hours and to the full availability. In this example, the service times are 10, 10, 5, 9, 8, 6, 7, 7, 5, 6, the corresponding demands are 20, 18, 13, 20, 15, 18, 15, 18, 13, 14, and the time windows are [512, 632], [238, 418], [115, 235], [144, 324], [345, 465], [302, 422], [227, 287], [326, 386], [626, 686], [575, 695] for customers 1 through 10, respectively.



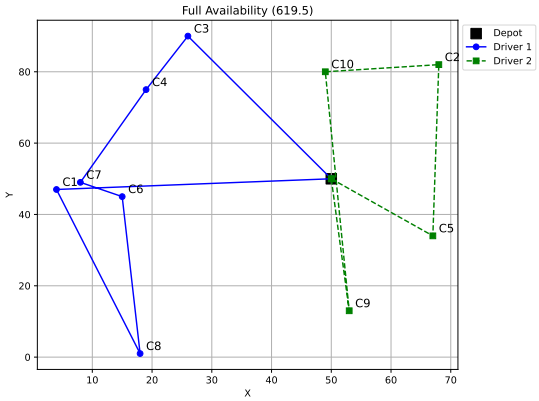
(a) Driver 1: Availability [120,360], Shift [120,300], Driver 2: Availability [240,480], Shift [240,450] Driver 4: Availability [480,720], Shift [480,630], Driver 3 and 5: not working.



(b) Driver 1: Availability [90,390], Shift [180,360], Driver 3: Availability [330,630], Shift [330,630], Driver 2, 4, and 5: not working.



(c) Driver 2: Availability [180,540], Shift [180,450], Driver 4: Availability [300,660], Shift [360,690], Driver 1, 3, and 5: not working.



(d) Full Availability [0, 720], Shifts: Driver 1: [180, 570], Driver 2: [330, 690], Driver 3, 4, and 5: not working.

Figure 3: Example: impact of driver availability on solution

Table 8: Example: impact of availability on cost structure.

	Availability: 4h	1h increase	2h increase	Full availability
Shift cost	445.0	125.0	155.5	187.5
Travel cost	137.5	255.0	386.0	432.0
3PL cost	95.0	285.0	95.0	0.0
Total cost	677.5	665.0	636.5	619.5

As availability increases in Figure 3, the cost structure shifts. In the 1-hour increase solution (665), Figure 3b, it is more cost-effective to assign customers to a 3PL rather than keeping three drivers, reducing shift costs (125) and travel costs (255), but increasing 3PL costs (285). In the 2-hour increase solution (636.5), as shown in Figure 3c, two drivers are utilized, with extended availability, which enables them to serve more customers, resulting in higher shift costs (155.5) and travel costs (386), but a decrease in 3PL costs (95). Finally, with full availability, total costs fall to 619.50, and all customers are assigned to drivers, resulting in longer routes and shifts. The cost gap between having 6 hours of availability and full availability (12 hours) is 2.7%. At this point, managers must decide whether to prioritize reducing total costs by encouraging longer driver availability or to accept shorter availability that gives drivers more flexibility, potentially increasing overall costs. Table 8 summarizes all the costs associated with this example.

6.2 Impact of working time limits

The second factor to consider for managers is the effect of shift duration on decision making. To examine the impact of relaxing these constraints, we first set driver availability to the planning horizon, allowing for full flexibility in scheduling.

Our findings in Table 9 suggest that removing shift duration limits, ranging from a minimum of 2 hours to a maximum of 8 hours, can yield several operational benefits, including

Table 9: Impact of ignoring shift duration on the average of some performance metrics

	Total cost	Shift cost	Travel cost	3PL cost	#Assigned drivers	Capacity utilization	Driver utilization
With shift limitation	684.08	201.69	428.64	53.75	12.52	0.40	0.43
Without shift limitation	673.54	202.13	423.43	47.98	12.57	0.43	0.46
Change%	-1.54	0.22	-1.22	-10.73	0.43	5.92	5.62

lower delivery costs, increased capacity utilization (defined as the ratio of used capacity to total capacity), improved driver utilization (the ratio of total assigned shifts to total availability), and reduced reliance on 3PL services. However, in many real-world contexts, shift durations are governed by labor agreements or government regulations, limiting managerial ability to implement such changes directly. Still, these results highlight the potential value of negotiating more flexible scheduling frameworks where feasible, such as through union discussions, pilot programs with voluntary overtime, or flexible shift bidding systems. Even incremental flexibility, when aligned with regulatory frameworks, could yield cost savings and improve internal resource use, ultimately reducing reliance on expensive outsourcing.

6.3 Impact of 3PL costs on customer and driver assignments

Another key concern for managers is how 3PL costs impact decision-making. To analyze this, we set f to five different values: 0.5, 0.9, 1, 1.1, and 1.5 and test some performance metrics shown in Figure 4. Our results reveal a clear cost-sensitivity threshold that can guide managers in outsourcing decisions.

When 3PL costs are relatively low ($f \leq 0.9$), outsourcing offers a cost-effective way to reduce reliance on in-house resources. This strategy can help companies scale rapidly without expanding their fleet. However, once 3PL costs approach or exceed parity with internal delivery costs ($f \geq 1$), the benefit of outsourcing diminishes sharply. Companies then stabilize their operations by maintaining a core team of in-house drivers and only

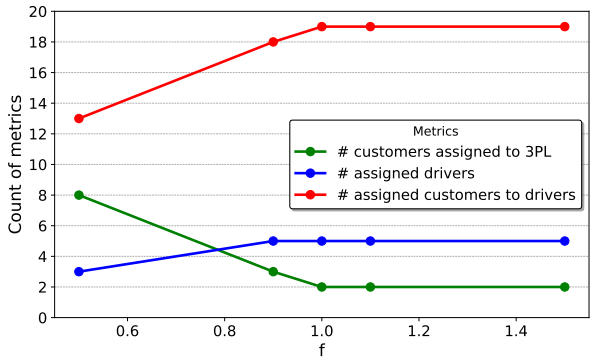


Figure 4: Impact of 3PL cost on different metrics.

outsourcing a minimal number of deliveries. This plateau effect implies that beyond a certain cost point, further increases in 3PL rates do not lead to further operational changes, suggesting a strategic cutoff beyond which renegotiating 3PL contracts or investing in internal capacity becomes more attractive.

7 Conclusions

In this study, we addressed the Vehicle Routing Problem with Driver Scheduling (VRPDS), which integrates vehicle routing with the workforce scheduling problem. We considered drivers' availability and formulated the problem as a deterministic model to minimize total costs, which include travel costs, shift costs, and the cost of serving customers by the 3PL. An ILS algorithm combined with a reassignment procedure was proposed to solve the VRPDS. Through computational experiments, we demonstrated that only a small number of instances were solved optimally using a commercial solver, while the ILS proved more efficient for larger instances.

This study highlights how adjusting policies related to driver availability, 3PL costs, and shift duration can influence operational costs and resource allocation. Relaxing driver availability lowers total costs by decreasing 3PL and traveling costs while increasing shift

costs. Higher 3PL costs result in more customers being assigned to in-house drivers, while loosening shift duration constraints reduces overall costs and leads to higher driver and capacity utilization. These insights provide managers with a pathway to optimize flexibility, cost control, and resource utilization.

Acknowledgments

This work was partly supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grants 2021-04037 and 2025-04195. We thank Compute Canada for providing high-performance parallel computing facilities.

Declaration of generative AI in scientific writing

During the preparation of this work, the authors used Grammarly and ChatGPT to edit the text. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

Uwe Aickelin and Kathryn A. Dowsland. An indirect genetic algorithm for a nurse-scheduling problem. *Computers & Operations Research*, 31(5):761–778, 2004.

Aliaa Alnaggar, Fatma Gzara, and James H. Bookbinder. Crowdsourced delivery: A review of platforms and academic literature. *Omega*, 98:102139, 2021.

Miguel Barbosa, João Pedro Pedroso, and Ana Viana. A data-driven compensation scheme for last-mile delivery with crowdsourcing. *Computers & Operations Research*, 150: 106059, 2023.

Jonathan F. Bard, Canan Binici, and Anura H. desilva. Staff scheduling at the united states postal service. *Computers & Operations Research*, 30(5):745–771, 2003.

Adam Behrendt, Martin W.P. Savelsbergh, and He Wang. A prescriptive machine learning method for courier scheduling on crowdsourced delivery platforms. *Transportation Science*, 57(4):889–907, 2023.

Jens O. Brunner, Jonathan F. Bard, and Rainer Kolisch. Flexible shift scheduling of physicians. *Health care management science*, 12(3):285–305, 2009.

Jean-François Cordeau, Gilbert Laporte, and Anne Mercier. Improved tabu search algorithm for the handling of route duration constraints in vehicle routing problems with time windows. *Journal of the Operational Research Society*, 55(5):542–546, 2004.

Philippe De Bruecker, Jeroen Beliën, Liesje De Boeck, Simon De Jaeger, and Erik Demeulemeester. A model enhancement approach for optimizing the integrated shift scheduling and vehicle routing problem in waste collection. *European Journal of Operational Research*, 266(1):278–290, 2018.

Andreas T. Ernst, Houyuan Jiang, Mohan Krishnamoorthy, and David Sier. Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, 153(1):3–27, 2004.

Nikolaus Frohner and Günther R. Raidl. A double-horizon approach to a purely dynamic and stochastic vehicle routing problem with delivery deadlines and shift flexibility. In *Proceedings of the 13th International Conference on the Practice and Theory of Automated Timetabling-PATAT*, volume 1, 2021.

Grand View Research. Canada last mile delivery market size & outlook, 2023–2030. `https://www.grandviewresearch.com/horizon/outlook/last-mile-delivery-market/canada`, 2023. Accessed: 2025-10-09.

Florian Grenouilleau, Antoine Legrain, Nadia Lahrichi, and Louis-Martin Rousseau. A set partitioning heuristic for the home health care routing and scheduling problem. *European Journal of Operational Research*, 275(1):295–303, 2019.

Chris Groër, Bruce Golden, and Edward Wasil. A library of local search heuristics for the vehicle routing problem. *Mathematical Programming Computation*, 2:79–101, 2010.

Walter J. Gutjahr and Marion S. Rauner. An ACO algorithm for a dynamic regional nurse-scheduling problem in Austria. *Computers & Operations Research*, 34(3):642–666, 2007.

Fang He and Rong Qu. A constraint programming based column generation approach to nurse rostering problems. *Computers & Operations Research*, 39(12):3331–3343, 2012.

Harold W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, 52(1):7–21, 2004.

Jenifer Lee. Advancements in last-mile delivery transforming Canada. `https://www.macmillanscg.com/blog/advancements-in-last-mile-delivery/`, 2025. Accessed: 2025-10-09.

Sheng Liu and Zhixing Luo. On-demand delivery from stores: Dynamic dispatching and routing with random demand. *Manufacturing & Service Operations Management*, 25 (2):595–612, 2023.

Helena R. Lourenço, Olivier C. Martin, and Thomas Stützle. Iterated local search. In *Handbook of Metaheuristics*, pages 320–353. Springer, 2003.

Simona Mancini and Margaretha Gansterer. Bundle generation for the vehicle routing problem with occasional drivers and time windows. *Flexible Services and Manufacturing Journal*, pages 1–33, 2024.

Simona Mancini, Margaretha Gansterer, and Richard F. Hartl. The collaborative consistent vehicle routing problem with workload balance. *European Journal of Operational Research*, 293(3):955–965, 2021.

Minakshi Punam Mandal, Alberto Santini, and Claudia Archetti. Tactical workforce sizing and scheduling decisions for last-mile delivery. *European Journal of Operational Research*, 323(1):153–169, 2025.

Vinícius R. Maximo, Jean-François Cordeau, and Mariá C.V. Nascimento. AILS-II: An adaptive iterated local search heuristic for the large-scale capacitated vehicle routing problem. *INFORMS Journal on Computing*, 36(4):974–986, 2024.

Jean-Yves Potvin and Jean-Marc Rousseau. An exchange heuristic for routeing problems with time windows. *Journal of the Operational Research Society*, 46(12):1433–1446, 1995.

Monia Rekik, Jean-François Cordeau, and François Soumis. Implicit shift scheduling with multiple breaks and work stretch duration restrictions. *Journal of Scheduling*, 13(1): 49–75, 2010.

Yingtao Ren, Maged Dessouky, and Fernando Ordóñez. The multi-shift vehicle routing problem with overtime. *Computers & Operations Research*, 37(11):1987–1998, 2010.

Martin W.P. Savelsbergh. The vehicle routing problem with time windows: Minimizing route duration. *ORSA Journal on Computing*, 4(2):146–154, 1992.

Anand Subramanian, Puca Huachi Vaz Penna, Eduardo Uchoa, and Luiz Satoru Ochi. A hybrid algorithm for the heterogeneous fleet vehicle routing problem. *European Journal of Operational Research*, 221(2):285–295, 2012.

Éric Taillard, Philippe Badeau, Michel Gendreau, François Guertin, and Jean-Yves Potvin. A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science*, 31(2):170–186, 1997.

The Conference Board of Canada. The outlook for Canada's transportation sector 2020-2040 (Post-COVID-19). Technical Report, Transport Canada, 2021. URL

https://publications.gc.ca/collections/collection_2021/tc/T22-250-2021-eng.pdf. Accessed: 2025-10-09.

Marlin W. Ulmer and Martin W.P. Savelsbergh. Workforce scheduling in the era of crowdsourced delivery. *Transportation Science*, 54(4):1113–1133, 2020.

U.S. Bureau of Labor Statistics. Occupational employment and wages, may 2023: 53-3031 driver/sales workers. https://www.bls.gov/oes/2023/may/oes533031.htm, 2024. Accessed: June 9, 2025.

Jorne Van den Bergh, Jeroen Beliën, Philippe De Bruecker, Erik Demeulemeester, and Liesje De Boeck. Personnel scheduling: A literature review. *European Journal of Operational Research*, 226(3):367–385, 2013.

Wenshu Wang, Kexin Xie, Siqi Guo, Weixing Li, Fan Xiao, and Zhe Liang. A shift-based model to solve the integrated staff rostering and task assignment problem with real-world requirements. *European Journal of Operational Research*, 310(1):360–378, 2023.

Dingtong Yang, Michael F. Hyland, and R Jayakrishnan. Tackling the crowdsourced shared-trip delivery problem at scale with a novel decomposition heuristic. *Transportation Research Part E: Logistics and Transportation Review*, 188:103633, 2024.

Zongcheng Zhang, Maoliang Ran, Yanru Chen, MIM Wahab, Mujin Gao, and Yangsheng Jiang. Dynamic crowdsourcing problem in urban–rural distribution using the learning-based approach. *Computers & Operations Research*, page 107292, 2025.