Journal of Cleaner Production 519 (2025) 145855

Contents lists available at ScienceDirect

Journal of Cleaner Production

journal homepage: www.elsevier.com/locate/jclepro

A green multi-period request assignment problem for road freight transport

Elham Jelodari Mamaghani [a,b], Yousef Ghiami [c,*], Emrah Demir [d], Tom Van Woensel [e]

[a] *Logistics and Optimization of Industrial Systems Laboratory of Computer Science, University of Technology of Troyes, Troyes, France*
[b] *Institute of Sustainable Business and Organizations Sciences and Humanities Confluence Research Center, UCLY, ESDES, France*
[c] *Department of Operations Analytics, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands*
[d] *Business & Economics Artificial Intelligence Research Network, Cardiff Business School, Cardiff University, Cardiff, UK*
[e] *School of Industrial Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands*

ARTICLE INFO

Handling Editor: Xin Tong

Keywords:
Green logistics
Pickup and delivery problem
Multi period planning
Bi-objective optimization
Reserved and selective requests

ABSTRACT

This paper addresses a multi-period pickup and delivery problem with time windows, where carriers must fulfill committed transport requests while deciding whether to accept additional requests to enhance their financial and environmental performance. Given the increasing focus on sustainability, the objective is to balance profitability and CO_2e emissions. To tackle this bi-objective problem, we propose a mixed-integer linear programming formulation that accounts for heterogeneous vehicles and both hard and soft time windows. To efficiently solve large-scale instances, we introduce a Hybrid Adaptive Large Neighborhood Search (HALNS) algorithm, which integrates population-based Tabu Search with a mutation operator within an ALNS framework. The proposed HALNS is benchmarked against multiple existing methods to assess its effectiveness and efficiency. Computational experiments demonstrate that HALNS efficiently solves large-scale instances, outperforming existing approaches. In addition, our numerical analysis provides key managerial insights for companies that want to achieve environmentally sustainable transport operations. Our numerical results indicate that imposing stricter emission targets can reduce CO_2e emissions by up to 40% while decreasing profits by approximately 21%. In contrast, increasing the size of the fleet leads to an increase in profits 15% and improves the performance of the delivery, but at the cost of higher emissions. Furthermore, relaxing the time window constraints improves operational flexibility, resulting in an increase in average profits of 5% while reducing emissions by approximately 7%. These findings highlight the trade-offs involved in sustainable logistics planning and offer actionable insights for managers.

1. Introduction

In today's interconnected supply networks, businesses have shifted their priorities beyond mere economic factors. Driven by environmental legislation and growing consumer awareness, there is a strong emphasis on decarbonizing transportation practices. Logistics service providers (LSPs) and retailers, among others, recognize that their transportation service choices would directly impact their perceived performance along sustainability indicators. As a result, these companies are actively looking for innovative ways to adopt cleaner vehicle technologies and improve their operational practices.

One of the most used delivery services is pickup and delivery (PD) in the less-than-truckload (LTL) sector. In this type of service, multiple requests should be transported, and each request involves collecting a load from a pickup point and delivering it to a designated drop-off point (Vaziri et al., 2019). Given that pickup and delivery points are generally unique for each request, these operations are highly prone to inefficiencies, e.g. empty hauls or long routes. This means that carriers active in this sector can play a key role in the decarbonization of transportation operations by using advance decision-making tools and reduce such inefficiencies, and increase utilization rate of their capacities.

The general Pickup and Delivery problem (PDP) was first formalized by Savelsbergh and Sol (1995) and has been widely studied in the literature since then, see, e.g., Ropke and Pisinger (2006), Xue (2022), Lyu and Yu (2023), Du et al. (2023), and Galiullina et al. (2024). Considering its wide range of applications in practice, numerous PDP variants have been proposed and studied. The most extensively studied variant of the PDP is the version with time windows (PDPTW), which requires the fulfillment of customers' orders within specific time windows, see, e.g., Ropke and Pisinger (2006), Al Chami et al. (2019), Sartori and Buriol (2020), Zhang et al. (2023) and Zhao et al. (2023). Due to its

* Corresponding author.
E-mail addresses: ejelodarimamaghani@univ-catholyon.fr (E. Jelodari Mamaghani), y.ghiami@vu.nl (Y. Ghiami), demire@cardiff.ac.uk (E. Demir), T.v.Woensel@tue.nl (T.V. Woensel).

https://doi.org/10.1016/j.jclepro.2025.145855
Received 14 June 2024; Received in revised form 3 May 2025; Accepted 27 May 2025
Available online 19 June 2025

computational complexity, the main focus of the research conducted on PDPTW has been the development of approximation algorithms, i.e., heuristics and metaheuristics (Sartori and Buriol, 2020; Wolfinger, 2021; Yang and Li, 2023; Zhao et al., 2023; Meng et al., 2024).

In transportation planning, a shipper requires a carrier to transport a set of requests. These requests can be categorized into two groups based on their flexibility in delivery time. The first group consists of *reserved requests*, which have a fixed delivery date. This means that when the shipper offers a reserved request, such as $R1$, to a carrier, they agree on a specific delivery date, say day t, on which $R1$ must be delivered. Consequently, $R1$ cannot be delivered on any other day, such as $t + 1$ or $t - 1$. The second group consists of *selective requests*, which offer more flexibility in the delivery timing. Instead of a fixed date, the shipper allows the carrier to deliver a selective request within a given period window. For example, the shipper can make a selective request $S1$ to the carrier and specify that it can be delivered on day t, $t + 1$, or $t + 2$. This flexibility enables carriers to adjust delivery schedules and shift selective requests on different dates to optimize capacity utilization and improve profitability, unlike reserved requests, which must be delivered strictly on their assigned date (Ben-Said et al., 2022). Generally, the allocation of selective requests to carriers is determined based on the combined profit of the carriers and the resulting impact on the customers, i.e., delivery time.

A group of researchers investigated PDPTW models with variable travel times aiming at profit maximization; see, e.g., Vaziri et al. (2019) and Sun et al. (2020). In the former study, a bi-objective PDPTW model was developed to minimize GHG emissions and maximize profit. The latter proposed a single-objective PDPTW model along with an ALNS algorithm as the solution method. In a recent study, Yang and Li (2025) examined the PDP in a single-period setting with time windows and handling time in the context of last-mile delivery. Real-life optimization problems require complex decision-making processes to find an optimal balance among several conflicting objectives (Evans, 1984). In their study, Soleimani et al. (2018) focused on green PDP (GPDP) for distributing new and re-manufactured products. They observed that significant cost reductions and environmental benefits can be achieved through the proposed multi-objective non-linear programming model. In another study, Christiaens and Vanden Berghe (2020) demonstrated the effectiveness of their algorithm in minimizing both travel costs and the number of utilized vehicles. Wang et al. (2023a) proposed a bi-objective optimization approach for a pickup and delivery problem within a two-echelon multi-depot multi-period location-routing framework by integrating vehicle sharing, and advanced metaheuristic optimization techniques. In a recent study, Santiyuda et al. (2024) investigated the bi-objective (i.e., minimization of costs and time) time-dependent PDP using learning-based optimization approaches.

Taking into account the increased awareness about sustainability constructs (environmental, social, and economic) and the need for immediate actions, "reducing costs" and "shortening delivery times" are no longer the only objectives that logistics companies are trying to achieve (Demir et al., 2022). Given the nature of the transportation industry, this sector contributes significantly to climate change, primarily through greenhouse gas (GHG) emissions due to its dependence on fossil fuels (McKinnon et al., 2015; Madankumar and Rajendran, 2018). The emergence of environmental concerns has also motivated researchers to focus on the emissions generated during transportation operations (Wang et al., 2018; Sun et al., 2019; Zhao et al., 2023). Soysal et al. (2018), for instance, investigated an environmentally-friendly Pickup and Delivery Problem with Time Windows (PDPTW), incorporating explicit fuel consumption, variable vehicle speed, and road categorization. Furthermore, Wang et al. (2020) explored multi-echelon transport systems within green logistics frameworks. These studies collectively contribute to the broader goal of sustainable urban logistics, aligning with ongoing research efforts to develop eco-efficient and cost-effective transportation networks. For a comprehensive overview of research conducted on green logistics, interested readers are referred to Demir et al. (2014b) and Moghdani et al. (2021).

Although there are many studies focusing on PDPTWs, most research overlooks key operational requirements. In practice, the transport services provided by logistics companies operate mainly on a rolling horizon. This means that for a carrier, it is important to know how its capacity is allocated on a specific day and how many requests are not planned for that day, so that they can be considered when devising the transportation plan for the upcoming days. However, most studies, if not all, focus only on single-period transport planning. Such models are indeed helpful for generating insights for practitioners and understanding the behavior of the corresponding transportation settings, such as specific trade-offs in a pickup and delivery setting. Nevertheless, they do not address the challenge that practitioners face, which is multi-period planning. Table 1 illustrates the differences between single- and multi-period settings. In this example, there are two trucks, each with a capacity for two requests. We assume that at the beginning of the first period, the planner faces a demand of three selective requests and three reserved requests. In a single-period setting, since there is no capacity planning for future periods, requests $S2$ and $S3$ are lost. This means that the shipper will not wait for another time period, as the same situation may occur again. This leads to inefficiencies in transportation operations, as the planner misses the opportunity to utilize the available capacity in future periods. In contrast, if the planner adopts a multi-period model, it can accept the selective requests at the beginning of the first period and deliver them in subsequent periods. By doing so, it not only improves capacity utilization, but also serves more requests, ultimately increasing revenues.

Logistics companies need to consider multiple objectives when planning their operations. Ben-Said et al. (2022) studied a bi-objective PDPTW with selective requests that considers maximization of profits while minimizing travel costs. In another study, PDP with hard time windows and time-dependent travel times was studied by Wang et al. (2023b). In a more recent study, Santiyuda et al. (2024) investigated a bi-objective PDPTW based on time-dependent travel times.

In this paper, we analyze a green multi-period request assignment problem in road freight transport (GMP-RAP) for less-than-truckload freight operations. Moreover, we assess the performance of this variant of PDPTW based on two main criteria, namely profit and CO_2e emissions. In GMP-RAP, each carrier has a set of reserved requests (mandatory service) and can handle additional selective requests from other carriers. Each request involves pickup and delivery operations, associated with origin, destination, quantity, two time windows, and profit. In this problem, the revenues generated from reserved requests are fixed whereas the revenues obtained from selective requests depend on the timing of the delivery operations. The second objective function serves to incorporate environmental considerations by reducing the production of CO_2e emissions. When addressing a bi-objective problem, a set of non-dominated solutions is specified, known as the Pareto front, allowing the decision-maker to express preferences. To solve GMP-RAP, we propose a hybrid adaptive large neighborhood search (HALNS) method, which is a three-phase algorithm consisting of an adaptive large neighborhood search (ALNS) algorithm, local search (for intensification), and population-based Tabu Search algorithm (TS_{pop}) (for diversification) with a mutation. The bi-objective problem is solved using ALNS after parameter tuning with Taguchi approach, followed by four local search methods to further improve the results. The solutions obtained are then fed into a population-based Tabu Search algorithm to enhance diversification within the solution space.

The main contributions of this paper is three-fold. First, this paper extends the PDP by incorporating a multi-period setting, addressing the complexities of real-world logistics operations over multiple time periods. Unlike traditional single-period models, our approach considers interdependencies between different time windows. Second, we introduce a bi-objective optimization framework that explicitly balances

Table 1
Single-period vs. multi-period planning.

Scenario	Requests available	Truck assignments	Remaining requests
Scenario 1: Repeated single-period planning (short-sighted)			
Period 1	R1, R2, R3, S1, S2, S3	(R1, R2), (R3, S1)	S2, S3 (Lost)
Period 2	R4 + (S2, S3)	(R4, Empty)	
Period 3	R5	(R5, Empty)	
Scenario 2: Multi-period planning (strategic)			
Period 1	R1, R2, R3, S1, S2, S3	(R1, R2), (R3, S1)	S2, S3 (Scheduled for later)
Period 2	R4 + S2 (carried over)	(R4, S2)	
Period 3	R5 + S3 (carried over)	(R5, S3)	

cost and environmental impact. By integrating emissions reduction into the decision-making process, our model provides a more sustainable approach to logistics, addressing the growing need for environmentally conscious transportation strategies. In addition, we incorporate realistic operational constraints, such as limited vehicle capacity and dynamic route adjustments, to improve the practical applicability of our method. Third, we conduct an extensive computational analysis using benchmark instances to validate the effectiveness of our approach. The results offer valuable managerial insights, illustrating the trade-offs between operational cost savings and sustainability goals in multi-period logistics planning. This study not only advances theoretical modeling, but also provides actionable strategies for logistics professionals seeking to optimize both efficiency and environmental impact.

The remainder of this paper is organized as follows. Section 2 focuses on the description of the problem and the resulting mathematical model. The solution methodology is provided in Section 3. Section 4 is devoted to computational experiments, and in Section 5 managerial insights are discussed. Finally, Section 6 summarizes the concluding remarks.

2. Problem description and mathematical formulation

The GMP-RAP consists of two objective functions. The economic aspect is included through the maximization of total profits and the environmental aspect is considered through the minimization of GHG emissions.

We now present the details of the GMP-RAP in which each service (i.e., pickup or delivery) is constrained with hard time windows. In this problem, a number of independent small road freight carriers form an alliance and aim to serve a total number of n requests. Within this context, there are two distinct types of requests: reserved and selective requests. The key difference between these two categories lies in the flexibility of the delivery date. Reserved requests must be served during a pre-defined period (i.e., a specific day), whereas selective requests come with the flexibility of being served within a given period. The allocation of selective requests is optimized to ensure fair distribution of profits and a reduction in GHG emissions. The GMP-RAP is defined on a directed graph $\mathcal{G} = (\mathcal{O}, \mathcal{E})$, where $\mathcal{O} = \{0, \ldots, 2n+1\}$ and $\mathcal{E} = \{(i,j), \forall i,j \in \mathcal{O}; i \neq j\}$ respectively represent the node and arc sets. In this graph, nodes 0 and $\{2n+1\}$ represent the depot. To meet transportation demand, a set of heterogeneous vehicles is considered, where each vehicle $k \in \mathcal{V} = \{1, 2, \ldots, v\}$ has a capacity of f_k and is available on certain days. During the planning horizon, if assigned, a vehicle starts at node 0 and ends its journey at node $2n+1$ after serving one or more requests.

In our problem setting, request $r \in \mathcal{R}$ is associated with two nodes; a pickup node $r \in \mathcal{P}$ and a delivery node $(n+r) \in \mathcal{D}$. We use d_r to denote the quantity associated with the request $r \in \mathcal{R}$, i.e. the quantity that should be loaded at the pickup node when serving a request. Since we do not consider split deliveries, the quantity loaded for request r should be fully unloaded at the delivery node $n+r$. This indicates that $d_{n+r} = -d_r$. The node $i \in \mathcal{O}$ is preferably accessible for pickup and delivery operations only during the time window $[a_i, b_i]$. Each reserved request has a specific period for pickup and delivery operations. In contrast, for the selective request $r \in \mathcal{R}^s$, carriers can choose from a set of days to deliver the request, which is called a period window $[\alpha_r, \beta_r]$. The period window of pickup node is the same as that of delivery node.We assume that transportation along the arc $(i,j) \in \mathcal{E}$ incurs a cost of c_{ij} and that each vehicle emits a specific amount of CO_2e emissions per km. The shipper, as the customer for the transportation service, pays an amount of m_r to deliver the request r. In short, we have the following assumptions: (i) each service (pickup or delivery) is constrained by hard time windows; (ii) there are two types of requests: reserved and selective. Reserved requests must be served during a pre-defined period. Selective requests can be served within a given period window; (iii) allocation of selective requests is based on the resulting profits and emissions.

The problem at hand aims to maximize profits and minimize the generated emissions. The profit function considers the revenues, costs, and rewards that carriers may receive for delivering a selective request earlier within the corresponding period window. The second objective is to minimize the total CO_2e emissions produced by the transportation service. Table 2 lists all the notation used in the mathematical model.

Maximize

$$\sum_{r_{pd}\in\mathcal{R}}\sum_{k\in\mathcal{V}}\sum_{h\in\mathcal{T}} m_{r_{pd}} y_{kh}^{r_{pd}} - \sum_{j\in\mathcal{P}}\sum_{k\in\mathcal{V}}\sum_{h\in\mathcal{T}} l_k x_{0jh}^k - \sum_{i\in\mathcal{O}}\sum_{j\in\mathcal{O}}\sum_{k\in\mathcal{V}}\sum_{h\in\mathcal{T}} c_{ij} x_{ijh}^k + \lambda \sum_{r_{pd}\in\mathcal{R}^s}\sum_{k\in\mathcal{V}}\sum_{h\in[\alpha_{r_{pd}},\beta_{r_{pd}}]} m_{r_{pd}} \left[\beta_{r_{pd}} - \alpha_{r_{pd}} - max\{0, hy_{kh}^{r_{pd}} - \alpha_{r_{pd}}\}\right] \tag{1}$$

Minimize

$$\sum_{i\in\mathcal{O}}\sum_{j\in\mathcal{O}}\sum_{k\in\mathcal{V}}\sum_{h\in\mathcal{T}} cd_{ij} x_{ijh}^k q_{ih}^k g \tag{2}$$

Subject to:

$$\sum_{\substack{i\in\mathcal{O}\\ \omega\neq i}} x_{i\omega h}^k - \sum_{\substack{j\in\mathcal{O}\\ \omega\neq j}} x_{\omega jh}^k = 0 \qquad \forall \omega\in\mathcal{W}, \forall k\in\mathcal{V}, \forall h\in\mathcal{T} \tag{3}$$

$$\sum_{\substack{j\in\mathcal{P}\\ j\neq 0}} x_{0jh}^k \leq 1 \qquad \forall k\in\mathcal{V}, \forall h\in\mathcal{T} \tag{4}$$

$$\sum_{\substack{i\in D\\ i\neq 2n+1}} x_{i(2n+1)h}^k \leq 1 \qquad \forall k\in\mathcal{V}, \forall h\in\mathcal{T} \tag{5}$$

$$\sum_{\substack{j\in\mathcal{W}\\ j\neq r_{pd}}} x_{r_{pd}jh}^k = y_{kh}^{r_{pd}} \qquad \forall r_{pd}\in\mathcal{P}, \forall k\in\mathcal{V}, \forall h\in\mathcal{T} \tag{6}$$

$$\sum_{\substack{j\in\mathcal{W}\\ j\neq n+(r_{pd})}} x_{j(n+(r_{pd}))h}^k = y_{kh}^{r_{pd}} \qquad \forall r_{pd}\in\mathcal{P}, \forall k\in\mathcal{V}, \forall h\in\mathcal{T} \tag{7}$$

$$\sum_{k\in\mathcal{V}}\sum_{h\in\mathcal{T}} y_{kh}^{r_{pd}} = 1 \qquad \forall r_{pd}\in\mathcal{R}^z \tag{8}$$

$$\sum_{k\in\mathcal{V}}\sum_{h\in[\alpha_{r_{pd}},\beta_{r_{pd}}]} y_{kh}^{r_{pd}} \leq 1 \qquad \forall r_{pd}\in\mathcal{R}^s \tag{9}$$

$$\sum_{r_p\in\mathcal{P}} y_{kh}^{r_{pd}} \leq \zeta_{kh} \qquad \forall k\in\mathcal{V}, \forall h\in\mathcal{T} \tag{10}$$

$$u_{ih}^k + t_{i(n+i)} + e_i \leq u_{(n+i)h}^k \qquad \forall i\in\mathcal{P}, \forall k\in\mathcal{V}, \forall h\in\mathcal{T} \tag{11}$$

$$u_{jh}^k \geq u_{ih}^k + e_i + t_{ij} x_{ijh}^k$$

Table 2
Notations used.

Sets	
$\mathcal{P} = \{1, 2, \dots, n\}$	Set of all pickup nodes (reserve $\mathcal{PR}^z$ and selective $\mathcal{PR}^s$)
$\mathcal{D} = \{n+1, \dots, 2n\}$	Set of all delivery nodes (reserve $\mathcal{DR}^z$ and selective $\mathcal{DR}^s$)
$\mathcal{O} = \{0, \dots, 2n+1\}$	Set of nodes including the depots
$\mathcal{W} = \mathcal{O} \backslash \{0, 2n+1\}$	set of nodes excluding the depots
$\mathcal{T}$	Set of periods measured in days
$\mathcal{V}$	Set of vehicles
$\mathcal{R}^z = \mathcal{PR}^z \cup \mathcal{DR}^z$	Set of reserve requests which is included pickup points ($\mathcal{PR}^z$) and delivery points ($\mathcal{DR}^z$)
$\mathcal{R}^s = \mathcal{PR}^s \cup \mathcal{DR}^s$	Set of selective requests which is included pickup points ($\mathcal{PR}^s$) and delivery points ($\mathcal{DR}^s$)
$\mathcal{R} = \mathcal{R}^z \cup \mathcal{R}^s = \mathcal{P} \cup \mathcal{D}$	Set of selective $\mathcal{R}^s$ and reserve $\mathcal{R}^z$ requests (all requests)
Parameters	
f_k	Capacity of vehicle k
l_k	Fixed cost of vehicle k
ζ_{kh}	Availability of vehicle k in period h. This binary parameter equals to 1 if the vehicle k is available in period h.
r_{pd}	pickup or delivery nodes of a request $r_{pd} \in (\mathcal{P} \cup \mathcal{D})$.
$m_{r_{pd}}$	Allocated revenue for a pickup of request r ($r_{pd} \in \mathcal{P}$). If $r_{pd} \in \mathcal{D}$, then $m_{r_{pd}} = 0$.
$pd_{r_{pd}}$	Time period in which node $r_{pd} \in \mathcal{R}^z$ should be served
r_p	Pickup node of request $r \in \mathcal{R}$
r_d	Delivery node of request $r \in \mathcal{R}$
λ	Reward rate (%) for the early delivery of selective requests
Δ	Penalty cost for arriving time after or before the time window
g	Represents the emissions produced per kilometer per kilogram of vehicle load (g/km/kg)
$[a_i, b_i]$	Time windows (hours) of node $i \in \mathcal{O}$
$[\alpha_{r_{pd}}, \beta_{r_{pd}}]$	Period window (days) of selective request which is the same for its pickup and delivery nodes ($r \in \mathcal{R}^s$)
t_{ij}	Travel time (s) of a vehicle along arc $(i, j) \in \mathcal{E}$
$c_{ij} = t_{ij}$	Travel cost (Euro) along the arc $(i, j) \in \mathcal{E}$
cd_{ij}	Travel distance along the arc $(i, j) \in \mathcal{E}$ (km)
e_i	Service time at node $i \in \mathcal{O}$
d_p	Demand at pickup node $p \in \mathcal{P}$
$d_{p+n} = -d_p$	Demand at delivery node $(p+n) \in \mathcal{D}$
$d_0 = d_{2n+1} = 0$	Demand at the depot
M	A very large number
Decision variables	
u^k_{ih}	Arrival time of vehicle $k \in \mathcal{V}$ in period $h \in \mathcal{T}$ at node $i \in \mathcal{O}$
q^k_{ih}	Current load of vehicle $k \in \mathcal{V}$ when it leaves node $i \in \mathcal{O}$ in period $h \in \mathcal{T}$
x^k_{ijh}	1, if and only if node $i \in \mathcal{O}$ is visited before $j \in \mathcal{O}$ in period $h \in \mathcal{T}$ by vehicle $k \in \mathcal{V}$; 0, otherwise
$y^{r_{pd}}_{kh}$	1, if and only if node $r_{pd} \in (\mathcal{R} = (\mathcal{P} \cup \mathcal{D}) = \mathcal{R}^r \cup \mathcal{R}^s)$ is served by the vehicle $k \in \mathcal{V}$; in period $h \in \mathcal{T}$ 0, otherwise
Auxiliary variables	
$\eta^{r_{pd}}_{kh}$	Difference between the scheduled service time $h \in \mathcal{T}$ and the lower bound of the period window $[\alpha_{r_{pd}}, \beta_{r_{pd}}]$ for selective request $r_{pd} \in \mathcal{R}^s$ served by vehicle $k \in \mathcal{V}$.
θ^{kh}_{ij}	Load-dependent emissions for vehicle $k \in \mathcal{V}$ moving from node $i \in \mathcal{O}$ to $j \in \mathcal{O}$ in period $h \in \mathcal{T}$.
Γ^k_{jh}	Difference between the arrival time of vehicle $k \in \mathcal{V}$ at node $j \in \mathcal{W}$ and the upper bound of a soft time window in period $h \in \mathcal{T}$, when the vehicle is late.
γ^k_{jh}	Difference between the arrival time of vehicle $k \in \mathcal{V}$ at node $j \in \mathcal{W}$ and the lower bound of the soft time window in period $h \in \mathcal{T}$, when the vehicle is early.

$$- M(1 - x^k_{ijh}) \qquad \forall i \in \mathcal{O}, \forall j \in \mathcal{O}, i \neq j, \forall k \in \mathcal{V}, \forall h \in \mathcal{T} \tag{12}$$

$$a_i \leq u^k_{ih} \leq b_i \qquad \forall i \in \mathcal{O}, \forall k \in \mathcal{V}, \forall h \in \mathcal{T} \tag{13}$$

$$q^k_{jh} \geq q^k_{ih} + d_j - (f_k + d_j)(1 - x^k_{ijh}) \qquad \forall i \in \mathcal{O}, \forall j \in \mathcal{O}, j \neq i, \forall k \in \mathcal{V}, \forall h \in \mathcal{T} \tag{14}$$

$$q^k_{ih} \in \{\max\{0, d_i\}, \min\{f_k, f_k + d_i\}\} \qquad \forall i \in \mathcal{O}, \forall k \in \mathcal{V}, \forall h \in \mathcal{T} \tag{15}$$

$$x^k_{ijh} \in \{0, 1\} \qquad \forall i \in \mathcal{O}, \forall j \in \mathcal{O}, \forall k \in \mathcal{V}, \forall h \in \mathcal{T} \tag{16}$$

$$u^k_{ih} \geq 0 \qquad \forall i \in \mathcal{O}, \forall k \in \mathcal{V}, \forall h \in \mathcal{T} \tag{17}$$

$$y^{r_{pd}}_{kh} = 0 \qquad \forall r_{pd} \in \mathcal{R}^s, \forall k \in \mathcal{V}, \forall h \notin [\alpha_{r_{pd}}, \beta_{r_{pd}}] \tag{18}$$

$$y^{r_{pd}}_{kh} = 0 \qquad \forall r_{pd} \in \mathcal{R}^z, pd_{r_{pd}} \neq h, \forall k \in \mathcal{V}, \forall h \in \mathcal{T} \tag{19}$$

The objective function has two components. The first objective, defined in (1), seeks to maximize total profits. The first term of the objective function represents revenue, the second term accounts for the fixed costs associated with vehicle usage, and the third term corresponds to transportation costs. The final term in the objective function represents the incentive mechanism, which encourages the fulfillment of selective requests as early as possible within their designated time window. The second objective, defined in (2), aims to minimize the total emissions produced during transport. Degree constraints are presented in (3). Constraints (4) and (5) ensure that a vehicle starts its journey from the depot and ends up at the depot. Constraints (6) and (7) ensure that, if a request is fulfilled, its delivery node is visited after its pickup node by the same vehicle within the same period. Constraints (8) restrict the model to serve reserved requests in their pre-specified periods. Selective requests are served within a pre-specified period window. These requirements are enforced by constraints (9). Constraints (10) ensure that a request can be served by a certain vehicle during the period h if the vehicle is available during that same period. We use constraints (11) to ensure that if a request is served, its delivery point is visited, considering the service time at the pickup point plus the travel time. Constraints (12) guarantee that travel time is respected if two nodes are visited consecutively. Time window constraints at the pickup and delivery nodes are enforced by (13). Constraints (14)–(15) ensure that the outgoing payloads from different nodes are set correctly for each node's demand and vehicle's capacity. Finally, Constraints (16)–(19) guarantee integrality and non-negativity conditions for the decision variables.

2.1. Linearization of the GMP-RAP

We linearize the objective function presented in objective function (1) by adding the additional variable $\eta^{r_p}_{kh}$ and constraints (21) and (22) as follows.

$$\max\Big(\sum_{r_{pd}\in\mathcal{R}}\sum_{k\in\mathcal{V}}\sum_{h\in\mathcal{T}} m_{r_{pd}} y^{r_{pd}}_{kh} \sum_{j\in\mathcal{P}}\sum_{k\in\mathcal{V}}\sum_{h\in\mathcal{T}} l_k x^k_{0jh} - \sum_{i\in\mathcal{O}}\sum_{j\in\mathcal{O}}\sum_{k\in\mathcal{V}}\sum_{h\in\mathcal{T}} c_{ij} x^k_{ijh} + \lambda \sum_{r_{pd}\in\mathcal{R}^s}\sum_{k\in\mathcal{V}}\sum_{h\in[\alpha_{r_{pd}},\beta_{r_{pd}}]} m_{r_{pd}}\left[\beta_{r_{pd}} - \alpha_{r_{pd}} - \eta^{r_{pd}}_{kh}\right]\Big) \tag{20}$$

$$\eta^{r_{pd}}_{kh} \geq h y^{r_{pd}}_{kh} - \alpha_{r_{pd}} \qquad \forall r_{pd}\in\mathcal{R}^s, \forall k\in\mathcal{V}, \forall h\in\mathcal{T} \tag{21}$$

$$\eta^{r_{pd}}_{kh} \geq 0 \qquad \forall r_{pd}\in\mathcal{R}^s, \forall k\in\mathcal{V}, \forall h\in\mathcal{T} \tag{22}$$

We define variable θ^{kh}_{ij} and constraints (24)–(27) to linearize the second objective function as follows.

$$\min\Big(\sum_{i\in\mathcal{O}}\sum_{j\in\mathcal{O}}\sum_{k\in\mathcal{V}}\sum_{h\in\mathcal{T}} cd_{ij}\theta^k_{ijh} g\Big) \tag{23}$$

$$\theta^k_{ijh} \leq M x^k_{ijh} \qquad \forall i,j\in\mathcal{O}, \forall k\in\mathcal{V}, \forall h\in\mathcal{T} \tag{24}$$

$$\theta^k_{ijh} \leq q^k_{ih} \qquad \forall i,j\in\mathcal{O}, \forall k\in\mathcal{V}, \forall h\in\mathcal{T} \tag{25}$$

$$\theta^k_{ijh} \geq q^k_{ih} - (1 - x^k_{ijh})M \qquad \forall i,j\in\mathcal{O}, \forall k\in\mathcal{V}, \forall h\in\mathcal{T} \tag{26}$$

$$\theta^k_{ijh} \geq 0 \qquad \forall i,j\in\mathcal{O}, \forall k\in\mathcal{V}, \forall h\in\mathcal{T} \tag{27}$$

2.2. The GMP-RAP with soft time windows

We now present an alternative MILP formulation for the GMP-RAP with soft time windows. In practice, strict adherence to hard time windows where customers must be visited within specific time intervals can be challenging due to the factors such as traffic, weather conditions, or operational disruptions. In many cases, some flexibility is desirable while still aiming at minimizing the deviations from specified time windows. Soft time windows allow for slight deviations from the prescribed time windows, where penalties are imposed for violations but are not as restrictive as hard time window constraints. This flexibility is essential in scenarios where it is impractical to always meet exact time windows, such as in dynamic or congested environments. To this end, we introduce two new variables, Γ^k_{jh} and γ^k_{jh}, which measure the deviation between the visit time and the bounds of the corresponding time window when a customer is visited either before or after the time window. We modify the first objective function in (20) to account for the penalty cost incurred when time window constraints are violated. This adjustment allows for a balance between flexibility and operational efficiency. The second objective function, which is focused on the core operational goals, remains unchanged, ensuring that the main objectives of cost minimization and service level optimization are preserved.

$$\max\Big(\sum_{r_{pd}\in\mathcal{R}}\sum_{k\in\mathcal{V}}\sum_{h\in\mathcal{T}} m_{r_{pd}} y^{r_{pd}}_{kh} - \sum_{j\in\mathcal{P}}\sum_{k\in\mathcal{V}}\sum_{h\in\mathcal{T}} l_k x^k_{0jh} - \sum_{i\in\mathcal{O}}\sum_{j\in\mathcal{O}}\sum_{k\in\mathcal{V}}\sum_{h\in\mathcal{T}} c_{ij} x^k_{ijh} + \lambda \sum_{r_{pd}\in\mathcal{R}^s}\sum_{k\in\mathcal{V}}\sum_{h\in[\alpha_{r_{pd}},\beta_{r_{pd}}]} m_{r_{pd}}\left[\beta_{r_{pd}} - \alpha_{r_{pd}} - \eta^{r_{pd}}_{kh}\right] - \Delta\sum_{j\in\mathcal{W}}\sum_{k\in\mathcal{V}}\sum_{h\in\mathcal{T}}(\Gamma^k_{jh} + \gamma^k_{jh})\Big) \tag{28}$$

Additionally, the hard time window constraints presented in (13) are replaced by soft time window constraints, specifically constraints (29)–(32), which integrate these deviations into the model and enforce non-negative penalty costs for violating time windows. This formulation provides a practical solution that can be applied in settings where perfect adherence to time windows is not always possible, but still needs to be controlled to maintain service quality and operational efficiency.

$$\gamma^k_{jh} \geq a_j \sum_{i\in\mathcal{O}}\sum_{k\in\mathcal{V}} x^k_{ijh} - u^k_{jh} \qquad \forall j\in\mathcal{W}, \forall h\in\mathcal{T} \tag{29}$$

$$\Gamma^k_{jh} \geq u^k_{jh} - b_j \sum_{i\in\mathcal{O}}\sum_{k\in\mathcal{V}} x^k_{ijh} \qquad \forall j\in\mathcal{W}, \forall h\in\mathcal{T} \tag{30}$$

$$\gamma^k_{jh} \geq 0 \qquad \forall j\in\mathcal{W}, \forall k\in\mathcal{V}, \forall h\in\mathcal{T} \tag{31}$$

$$\Gamma^k_{jh} \geq 0 \qquad \forall j\in\mathcal{W}, \forall k\in\mathcal{V}, \forall h\in\mathcal{T}. \tag{32}$$

Given the NP-hard nature of the problem under investigation, we introduce a novel hybrid metaheuristics algorithm in the next section.

3. Solution method

This section proposes a hybrid solution approach that combines the ALNS and a population-based TS with a mutation operator (TS_{pop}) to solve the multi-objective problem. We name this hybrid algorithm as HALNS and use it to efficiently solve the GMP-RAP and produce high-quality solutions even for large-sized instances. Algorithm 1 presents the basic steps of HALNS. In addition to Algorithm 1, we provide the flowchart in Appendix A. In the following sections, we elaborate on the different steps of our solution methodology.

The solution method starts with the Randomized Clarke and Wright (RCW) algorithm and generates reasonably good initial feasible solutions. Subsequently, the ALNS algorithm, introduced by Ropke and Pisinger (2006), is used to explore the feasible region and generate multiple new and improved solutions. ALNS uses multiple removal and insertion operators. The newly generated solutions are then merged with a set of randomly generated solutions and provided as input to the TS_{pop} algorithm to identify high-quality solutions as the fine-tuning mechanism. The TS_{pop} algorithm also incorporates a mutation operator to avoid premature convergence to a suboptimal solution. This operator introduces controlled solution variations, ensuring that the search explores different facets of the solution space.

Algorithm 1: Pseudocode of HALNS algorithm.

```
procedure THE HYBRID ALNS ALGORITHM
    Initialize and tune the parameters of both ALNS and TS_pop algorithms
    Generate an initial solution using RCW algorithm (section 3.1)
    Execute ALNS algorithm using weighting method (section 3.2)
    Generate a set of random solutions
    Combine the solutions of the ALNS and the randomly generated ones
    Execute TS_pop on the population (section 3.3)
    Return final solution
```

3.1. An initial solution

In the proposed method, a solution is composed of three vectors, i.e., X, Y, and Z, respectively, present pickup and delivery points of requests, periods to serve requests, and vehicles to serve requests. The length of the vector X is equal to the number of pickup and delivery points, and the size of the other two vectors is the number of pickup points.

To obtain a feasible, high-quality initial solution, we use the RCW algorithm introduced by Prodhon (2011). Given the presence of selective requests, the algorithm is tailored to ensure that selective requests can be incorporated into the initial solution. To end this, the algorithm introduces a random element. When a randomly generated number exceeds a predefined threshold, selective requests are added to the initial solution, but only after all reserved requests are considered. In contrast, only reserved requests are included in the initial solution when the random number falls below the specified threshold.

The criterion based on which a request is chosen to be included in the initial solution is profitability, i.e., requests with higher profit are chosen first. When adding a request to the initial solution is feasible, its pickup and delivery points are inserted back into the same route. If adding a request makes the solution unfeasible, that request takes precedence, as the algorithm initiates the construction of a new route. Creating new routes continues until there are no more pending requests to be fulfilled.

3.2. ALNS algorithm

We use the weighted-sum method to consider both objectives (K_1 and K_2). More specifically, we assign a corresponding weight to each objective. In this problem, the first objective should be maximized, and the second one should be minimized. To consider both objective functions in the same objective function, we choose the following structure:

$$Min(-w_1 K_1 + w_2 K_2) \tag{33}$$

$$w_1 + w_2 = 1 \tag{34}$$

To better search for the feasible region, we apply two types of operators on the initial and each current solution: removal and insertion operators. At the beginning of the ALNS, all operators have equal *probability* to be selected. After a certain number of iterations (i.e., one segment), the probabilities are updated according to the operators' performance. We adopted the roulette wheel mechanism proposed by Ghiami et al. (2019) to select operators and update their scores. Before starting the next segment, the algorithm also performs a local search procedure to improve a current solution. The detailed steps of ALNS are given in Algorithm 2.

Algorithm 2: Pseudocode of ALNS algorithm

```
procedure ALNS Algorithm
    Select the best solution S_best as a initial solution
    Initialize the weights and scores of all operators
    In each segment
        Choose the policies for the selection of requests
        In each iteration
            Apply ALNS operators (Section 3.2.1 and Section 3.2.2 )
            If the solution is better than the last one
              Update performance score of the selected operator 3.2
            End if
        End of iteration
        Update the weights
        Call local search procedure (Section 3.2.3)
    End of segment
    Update the global best solution (S_best)
```

To have a population of solutions at the end of the ALNS process, rather than just a single solution at the end of each segment, we retain the feasible "current solution" after the end of fifty iterations. Consequently, after completion of the ALNS algorithm, rather than obtaining a single solution, we generate a population of solutions that are used during the subsequent phase when running the TS_{pop} algorithm.

3.2.1. Removal operators

Removal operators remove one or more requests from an existing solution. In each iteration, we generate a random number and, on the basis of its value, the algorithm chooses one of the removal operators. We employ two rules to remove a set of requests from a solution; either (i) both types of requests have an opportunity to be removed, or (ii) only selective requests can be removed from the solution. The selection between these two cases is made randomly.

We have adopted six removal operators from the literature and introduced three new ones (the last three in the following list). Each of these operators helps to increase the diversification or intensification of the solution method. In the following, we elaborate on the removal operators used in the solution method.

1. Random removal operator: It chooses a random set of requests to remove from a solution (Ghiami et al., 2019).

2. Least profit removal operator: This operator removes a random number of requests with the lowest profit (Li et al., 2016)

3. Least paid removal operator: This operator removes a group of requests that have the lowest revenue (Li et al., 2016).

4. Most expensive removal operator: The operator targets the most costly requests in a solution and removes them. (Li et al., 2016)

5. Shaw removal operator: This operator removes all requests similar to a seed request in terms of distance, time, and load (Shaw, 1998).

6. Similar price removal operator: This operator removes requests with similar prices (Li et al., 2016).

7. Highest emission removal operator: This operator removes a selective request from the list of requests based on the amount of CO_2e emissions. In other words, it targets and removes the selective request with the highest emissions.

8.Time removal operator: This operator removes selective requests from a route based on their arrival and departure times. For selective request i, two values are calculated: A_i, the absolute difference between the arrival time and the lower bound of the time window, and B_i, the absolute difference between the departure time and the upper bound. The selective request with the highest score $A_i + B_i$ is then removed from its route.

9. Longest period window removal operator: The operator removes a group of selective requests with the longest period windows.

3.2.2. Insertion operators

To build a new solution, we use a set of four insertion operators, out of which three are adopted from the literature. The algorithm generates a random number to choose one of the operators. Initially, these operators have the same chance of being chosen. Once an operator is chosen, the algorithm randomly selects a policy for insertion. For example, (i) does not distinguish between selective and reserved requests, or (ii) only profitable selective requests are inserted, and (iii) the priority is given to reserved requests. When the stopping criteria are met, the solution method checks the feasibility of the solution at hand. The solution method uses the following insertion operators:

1. Basic greedy insertion operator: This operator inserts each removed request into the best possible position in the solution (Ghiami et al., 2019).

2. 2-Regret insertion operator: This operator, first, finds the best and second best positions for each request in the removed list. Then, it finds the request with the highest difference between the best and second best position and inserts it into the best possible position (Demir et al., 2012).

3. 3-Regret insertion operator: This operator finds the removed request with the highest difference between its best, second, and third best positions, and inserts it in the best position (Parragh and Schmid, 2013).

4. Selective request insertion: This operator sorts the unfulfilled selected requests based on their profits and inserts the best one into the current solution.

3.2.3. Local search procedure

At the end of each segment, the algorithm randomly selects one of the following local moves to search for a better solution; (i) it randomly picks one of the selective requests and changes the order of either its pickup or delivery point in its route, (ii) it performs a similar move to the previous one but on a reserved request, (iii) the algorithm moves the positions of both the pickup and delivery points of a randomly selected request on its route, (iv) it randomly removes one of the selective requests from the solution. The new solution is accepted only if it improves the objective function.

3.3. Population-based tabu search algorithm with a mutation operator (TS_{pop})

Tabu search (TS) algorithm is proposed by Glover (1989), and more information on the algorithm can be found in Grendreau and Potvin (2010). Using different operators, the TS algorithm locally searches a neighborhood for potentially better solutions. We implement TS_{pop} on a population of solutions. We use the solutions generated by the ALNS and, in addition to that, we randomly produce a set of feasible solutions. Three Tabu lists are introduced and defined as vectors X, Y, and Z. Additionally, we benefit from three operators: swap, reversion, and reinsertion. The details of the TS_{pop} are provided in Algorithm 3.

Algorithm 3: Pseudocode of TS_{pop}.

```
procedure TS_pop ALGORITHM
    In each iteration
        While the number of examined operation < the number of operations on vector X (Y and Z)
            Choose an un-examined operation (from vectors X, Y, and Z) and obtain a new solution
            Update the current solution
            If the current solution is better than the best solution and the operation is not in tabu list
                Update the best solution and put the operation in the tabu list
            End if
        End while
        If the random number generated from [0,1] < mutation probability
            Execute mutation operator
        End if
        Update best solution by the current solution if the current solution has a better value
    End iteration
```

3.3.1. Diversification operators

A mutation operator is used to add diversification. Depending on the structure of the vectors, two mutation operators are implemented. The first operator is proposed for the vector X, and the second is developed for the vectors Y and Z.

Due to the structure of vector X (permutation structure), two elements are randomly selected at first. In the next step, one swap, reversion, or relocation operator is selected to apply to the two chosen elements. The mutation operator on vector Z (vector Y) is implemented in Algorithm 4.

The same approach is used to implement the mutation operator in the vector Y by adapting the algorithm. Each selected element of vector Z in Algorithm 4 is substituted with a random integer generated between one and the length of the period, inclusive. This dual mutation strategy improves the exploration of the solution space by introducing variability in both vector Y and vector Z. By diversifying the elements in these vectors, the algorithm increases its chances of escaping local optima and finding more robust solutions. Consequently, this process contributes to the overall effectiveness and efficiency of the HALNS algorithm.

Algorithm 4: Mutation operator on vector Z (and vector Y).

```
The number of mutation elements (e) are determined. The procedure is described in the following:
    A random integer number (r) is produced between 1 and the maximum number of vehicles.
    Mutation rate is considered as r' and multiplies r.
    Finally, round the value of r × r' to smallest integer greater than its value.
e elements of vector Z is selected (Same for vector Y).
e random integer number is produced between 1 and the maximum number of vehicles.
The elements produced in the previous step are replaced by the generated elements in the previous step.
```

4. Computational experiments

This section provides the detailed results obtained with the HALNS algorithm. We applied our solution algorithm to the benchmark instances used by Li et al. (2016). When possible, we compared the results of our algorithm to that of Li et al. (2016) based on four different metrics. The proposed algorithm is coded in MATLAB 2017 and has been used to solve 50 instances. The numerical experiments were performed on a laptop with an i5 core, 4 GB RAM, and 4.3 GHz speed. It is important to note that, for each instance, to ensure the robustness and reliability of our results, we conducted three independent runs. This approach helps mitigate the impact of problem size variations on algorithmic performance. The primary motivation for performing multiple runs is to reduce variability that may arise due to differences in instance structures, thereby providing more stable and generalizable results. Then, the average of these three runs is reported in this section. To this end, the instances are categorized into small, medium, and large categories. Small-size instances include 10 and 20 requests. Medium-size instances consist of instances with 30, 40, and 50 requests. Finally, the instances with 100 requests are classified as large-sized instances. In Table 3, we provide the details of all 50 instances.

In our experiments, each instance is identified as R-R^z-R^s, where R represents the total number of requests, R^z is the number of reserved requests and R^s describes the number of selective requests. Since we are utilizing benchmark instances from Li et al. (2016), designed as a single-period problem, we randomly determine the number of periods and set lower and upper bounds for selective requests' periods. Moreover, we keep the original values of the time windows. When multiple instances have the same number of selective and reserved requests, we differentiate them by appending a letter to their names. Notably, these instances differ in various attributes, including period windows, time windows, and vehicle capacities.

4.1. Results of the HALNS algorithm

In this section, we present a detailed analysis of the numerical experiments conducted using the HALNS algorithm. We employ the Taguchi approach to fine-tune parameters to ensure more robust results. Given the bi-objective nature of our problem, we focus on convergence and diversity as key performance measures for assessing solution quality. To quantify these aspects, we utilize the multi-objective coefficient of variation (MOCV), which integrates the Mean Ideal Distance (MID) and diversity metrics. These measures allow us to evaluate the trade-offs between solution proximity to the ideal point and solution spread along the Pareto frontier.

4.1.1. Parameter tuning with Taguchi approach

The Taguchi approach is a popular method for tuning parameters to have more robust results (see, e.g., Sels and Vanhoucke, 2012; Zarandi et al., 2013). Given that the problem is a bi-objective, convergence and diversity of solutions are the main performance measures to assess the

Table 3
The instances used and their respective numbers.

Instance	Number (#)	Vehicles (#)	Period (#)	Instance	Number (#)	Vehicles (#)	Period (#)
10-5-5a	1	3	3	40-20-20a	26	12	5
10-5-5b	2	3	3	40-20-20b	27	12	5
10-5-5c	3	3	3	40-20-20c	28	12	5
10-3-7d	4	3	3	40-15-25d	29	12	5
10-3-7e	5	3	3	40-15-25e	30	12	5
10-3-7f	6	3	3	40-15-25f	31	12	5
10-7-3g	7	3	3	40-25-15g	32	12	5
10-7-3h	8	3	3	40-25-15h	33	12	5
10-7-3i	9	3	3	40-25-15i	34	12	5
20-10-10a	10	6	5	50-25-25a	35	15	4
20-10-10b	11	6	5	50-25-25b	36	15	4
20-10-10c	12	6	5	50-25-25c	37	15	4
20-5-15d	13	6	5	50-20-30d	38	15	4
20-5-15e	14	6	5	50-20-30e	39	15	4
20-5-15f	15	6	5	50-20-30f	40	15	4
20-15-5g	16	6	5	50-30-20g	41	15	4
20-15-5h	17	6	5	50-30-20h	42	15	4
20-15-5i	18	6	5	50-30-20i	43	15	4
30-15-15c	19	9	4	100-50-50a	44	30	5
30-10-20d	20	9	4	100-50-50b	45	30	5
30-10-20e	21	9	4	100-50-50c	46	30	5
30-10-20f	22	9	4	100-25-75d	47	30	5
30-20-10g	23	9	4	100-25-75e	48	30	5
30-20-10h	24	9	4	100-25-75f	49	30	5
30-20-10i	25	9	4	100-75-25g	50	30	5

quality of the values assigned to parameters. We employ MOCV as our metric to evaluate both. MOCV considers both the MID and diversity, as proposed by Hajipour et al. (2016). In this study, we use the concept of diversity to quantify the dispersion of Pareto solutions, a concept elaborated in detail by Zitzler (1999) and Moradi et al. (2011).

Diversity: This metric evaluates the spread of solutions on the Pareto front obtained by the HALNS algorithm. It offers valuable insights into the extent and consistency of solutions along the established Pareto frontier. A higher diversity score indicates a more equitable distribution of solutions, implying that the algorithm has comprehensively explored a wide range of potential trade-offs. In contrast, a lower diversity score might suggest a clustering of solutions within particular areas of the Pareto front.

MID: The MID metric offers valuable insights into the proximity of solutions on the Pareto front to the ideal reference point. A lower MID value indicates that the solutions closely approximate the ideal point. In comparison, a higher MID value suggests that the solutions are distant from the ideal point, indicating potential room for improvement within the set of solutions. In summary, MID measures the alignment between solutions and the optimal reference, with lower values signifying superior performance.

The MOCV calculation determines the distance between each solution on the Pareto front and the reference point. This distance can be calculated using various metrics, including the Euclidean distance or the weighted Euclidean distance. The MOCV can be calculated, using the following formula $MOCV = \frac{MID}{Diversity}$.

To reduce the number of experiments, Taguchi proposes fractional factorial experiments (FFE) instead of full factorial designs, see, e.g., Sadeghi et al. (2014), and Allahyari et al. (2021). In this research, we follow the approach proposed by Allahyari et al. (2021) to tune the parameters. These parameters are used in multiple algorithms discussed in this research work. To assess the effectiveness of the HALNS developed in this paper, we benefit from several known algorithms, namely, the Non-Dominated Sorting Genetic Algorithm II (NSGA-II), Multi-Objective Memetic Algorithm (MOMA), Multi-Objective Simulated Annealing Algorithm (MOSA), and Multi-Objective Evolutionary Algorithm (MOEA). In total, we define five parameters for HALNS, four for NSGA-II, five for MOMA, four for MOEA, and three for MOSA. Each parameter is tested at three levels (low, medium, and high). Using the Taguchi method, we applied an L9 orthogonal array for all algorithms, resulting in 9 experimental runs per algorithm. We assume 100 segments for the HALNS, each including 150 iterations. When updating the score of each operator at the end of a segment, we consider an increment of 3 points. The final parameter settings are summarized in Table 4.

4.1.2. The effect of the TS_{pop} algorithm on the solution quality

This section investigates the effect of the TS_{pop} algorithm on the performance of the proposed solution method. We use a set of instances that contain a total number of 50 requests but are different in terms of one or more of the other features, e.g., the number of reserved and selective requests, the number of vehicles, their capacities, period windows, and time windows. To quantify the effect of ITS, we first run the HALNS algorithm on each instance three times and then exclude ITS from the algorithm and run the ALNS for three more rounds, C_i, $i = 1, 2, \dots, 6$. We then set C_{best} equal to the best value of the six outputs. To make the results comparable, we calculate the Ratio of Performance to Deviation (RPD) for each run as $RPD_i = 100 \times |\frac{C_{best} - C_i}{C_{best}}|$.

Table 5 summarizes the results of the analysis. The table shows that, on average, HALNS which benefits from the TS_{pop} algorithm performs better in the instances used in the experiments. In this table, *Diversity* is the spread of solutions on the Pareto front, *MID* stands for the proximity of solutions on the Pareto front to the ideal reference point, and *Time* is the total CPU time required to run an instance. When calculating RPD for each metric, a lower value is more desirable for this comparison.

As shown in Table 5, the TS_{pop} algorithm consistently improves the quality of the solution in all three metrics used in the experiments. In particular, notable improvements can be made for instances that feature both types of time windows when employing the ITS algorithm along with the ALNS heuristic in *Diversity* and *MID* metrics.

4.1.3. Performance assessment of the operators

To assess the operators' performance in the solution method, we choose the instance 30-15-15c with both soft and hard time windows. We run these two instances and measure the percentage of iterations that each operator is chosen in the removal or insertion stages. The result of this experiment is presented in Tables 6 and 7. As evident from these two tables, the proposed operators' performance, especially that of the highest emissions operator among the proposed removal operators and the insertion of selective requests, has been quite impressive.

Table 4
Parameters, their corresponding ranges, and final calibrated values of the parameters.

Method	Parameter	Range	Low	Medium	High	Calibrated values
HALNS	it (iteration)	50–250	50	150	250	150
	tl_x (Tabu list)	0.20–0.50	0.20	0.35	0.50	0.50
	tl_y (Tabu list)	0.10–0.40	0.10	0.25	0.40	0.25
	tl_z (Tabu list)	0.10–0.50	0.20	0.35	0.50	0.20
	mr (mutation probability)	0.24–0.30	0.24	0.27	0.30	0.30
NSGA-II	npop (size of population)	100–200	100	150	200	150
	it (maximum number of iterations)	200–600	200	400	600	60
	pc (crossover probability)	0.40–0.80	0.40	0.60	0.80	0.60
	pm (mutation probability)	0.05–0.30	0.05	0.18	0.30	0.05
MOMA	spop (size of population)	100–300	100	200	300	200
	nit (maximum number of iterations)	200–400	200	300	400	400
	cr (crossover rate)	0.40–0.80	0.40	0.60	0.80	0.60
	mr (mutation rate)	0.05–0.30	0.05	0.175	0.30	0.05
	lsf (local search frequency)	0.10–0.50	0.10	0.30	0.50	0.30
MOEA	nop (size of population)	100–300	100	200	300	100
	mit (maximum number of iterations)	100–500	100	300	500	500
	crr (crossover rate)	0.50–0.90	0.50	0.70	0.90	0.70
	mur (mutation rate)	0.05–0.30	0.05	0.175	0.30	0.175
MOSA	T_0 (initial temperature)	100–500	100	200	500	200
	mnit (maximum number of iterations)	100–500	100	300	500	500
	cr (cooling rate)	0.90–0.95	0.90	0.925	0.95	0.95

Table 5
Performance of ALNS and HALNS along the three metrics with hard and soft time windows.

Time window	Instance	Metric					
		Diversity		MID		Time (s)	
		HALNS	ALNS	HALNS	ALNS	HALNS	ALNS
Hard	50-25-25a	17.26	20.13	15.06	24.50	19.12	18.93
	50-25-25b	15.12	17.21	20.14	31.03	30.19	29.31
	50-25-25c	20.27	28.07	19.13	28.12	38.88	36.96
	50-20-30d	17.11	25.40	25.49	33.30	36.22	38.74
	50-20-30e	10.08	19.05	21.50	25.09	28.87	29.75
Soft	50-25-25a	18.22	33.67	14.11	15.57	30.83	30.94
	50-25-25b	13.20	23.15	18.20	24.16	36.91	36.10
	50-25-25c	22.64	29.20	19.66	23.72	32.32	33.15
	50-20-30d	13.65	21.48	27.58	30.40	36.70	35.68
	50-20-30e	19.33	31.21	25.37	32.74	32.81	36.69

4.2. HALNS vs. CPLEX

There are several approaches for solving bi-objective problems (Demir et al., 2014a). One method is to scalarize the objective functions. This approach converts a bi-objective problem into a single-objective problem. The weighting and ϵ-constraint methods are popular scalarization approaches to obtain high-quality Pareto solutions. In this research, we use the CPLEX solver to solve a set of small-sized instances using these methods. To evaluate the efficiency of the proposed HALNS, we also solve the same instances using the HALNS heuristic algorithm. The results of these analyses are illustrated in Table 8.

The results show that the HALNS heuristic algorithm runs faster than the exact method with either ϵ-constraint and weighting methods. Finding a feasible solution in one hour is impossible for larger instances, such as those with 20 requests. At the same time, the HALNS algorithm can converge on a high-quality solution in relatively shorter times. To compare the number of Pareto solutions found with each method, we present Fig. 1 to illustrate the Pareto fronts of 10-3-7e instance. The efficiency of these methods is assessed based on their ability to generate a well-distributed Pareto front with a reasonable computation time. By efficient, we refer to the method's ability to generate a more continuous and gradual trade-off between the two objectives (profit and CO_2e emissions), achieve better diversity on the Pareto front to offer decision makers a wider range of solutions, and improve convergence towards the true Pareto optimal front.

As seen from Fig. 1, the ϵ-constraints method provides a more gradual trade-off between profit and CO_2e emissions compared to the weighting method, which tends to cluster solutions. However, the HALNS algorithm outperforms both, producing a broader and denser Pareto front. HALNS consistently outperforms the other two methods in the number of solutions, diversity, and time metrics. Our key observations indicate that HALNS produces more Pareto solutions (a total of 25 solutions were obtained by HALNS, compared to 18 solutions with the ϵ-constraint method and 12 solutions with the weighting method), ensuring a better representation of trade-offs. The ϵ-constraint method offers smoother trade-offs than the weighting method but comes at a higher computational cost. In contrast, the weighting method results in fewer and more clustered solutions, limiting the decision-maker's flexibility. HALNS, however, balances computational efficiency with high-quality Pareto solutions, making it the preferred approach for large-scale instances. These findings reinforce that HALNS is more effective and computationally efficient than exact methods.

4.3. HALNS vs. ALNS

To the best of our knowledge, no advanced heuristics algorithm has been proposed to solve a multi-period, multi-objective PDP with heterogeneous fleet of vehicles. Therefore, to evaluate the performance of HALNS, we can only compare it with algorithm designed for single-objective and single-period settings with homogeneous vehicles. As a benchmark, we consider the algorithm proposed by Li et al. (2016). Since Li et al. (2016) focuses solely on profit maximization in a single-period setting with hard time windows and homogeneous vehicles, we modify our assumptions to ensure comparability. Specifically, we discard the objective on emissions, set the number of periods equal to one, and assume a set of a homogeneous fleet of vehicles. We then run the algorithm on a set of instances introduced by Li et al. (2016). Table 9 illustrates the results of this analysis in terms of profit values and CPU times.

As shown in Table 9, the proposed HALNS algorithm outperforms the algorithm proposed by Li et al. (2016) in most instances. We achieved higher profit values for all instances except '100-75-25g'. Additionally, we observed that the computational times for only a few instances were smaller than those of our algorithm. This represents a strong performance, especially considering that our algorithm was designed for a more complex and practical problem.

Table 6
Performance of the removal operators used in the HALNS algorithm.

Time window	Random removal (%)	Least profit (%)	Most expensive (%)	Shaw removal (%)	Similiar price (%)	*Highest emission* (%)	*Longest period* (%)	*Time removal* (%)
Hard	4.30	20	16.60	4.50	3.30	25	13	13.30
Soft	5.33	16.60	16.60	5.33	9	18.60	16.60	11.94

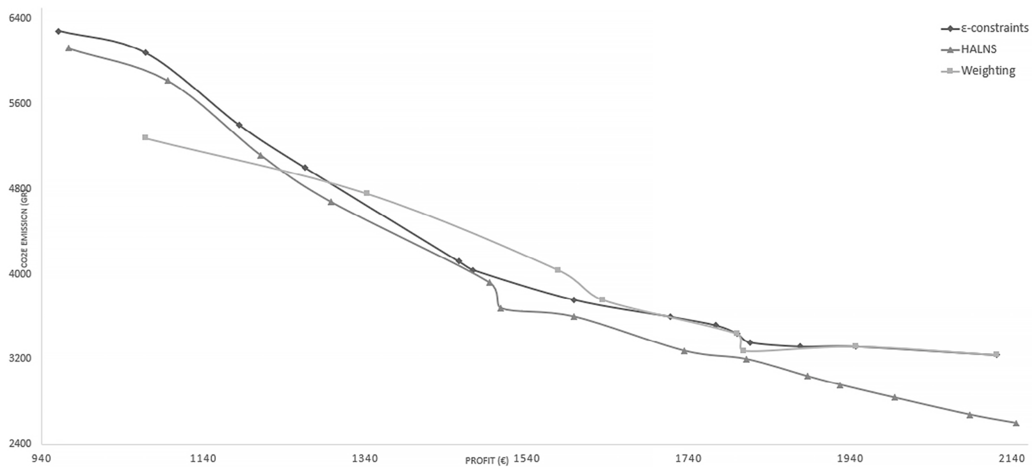


Fig. 1. Performance of the HALNS vs. the two exact methods.

Table 7
Performance of the insertion operators used in the HALNS.

Time window	Basic greedy (%)	2-regret (%)	3-regret (%)	*Selective requests* (%)
Hard	18.66	26.71	21.33	33.30
Soft	20	21.33	32.10	26.66

Table 8
A comparison of computational times on small-sized instances.

Time window	Instance	CPU time (s)		
		Exact solution		HALNS
		Weighting method	ϵ-constraint method	
	10-5-5a	162.14	158.44	23.18
	10-5-5b	183.16	185.31	25.55
	10-5-5c	188.20	189.09	20.05
	10-3-7d	204.14	201.19	21.37
Hard time windows	10-3-7e	187.75	176.25	20.87
	10-3-7f	205.70	202.59	19.33
	10-7-3g	196.10	190.66	20.88
	10-7-3h	195.88	182.60	21.48
	10-7-3i	208.13	194.12	21.51
	20-10-10a	3600	3600	31.23
	20-10-10b	3600	3600	32.75
	10-5-5a	167.50	115.55	22.20
	10-5-5b	182.12	211.67	23.68
	10-5-5c	188.90	194.37	21.10
	10-3-7d	180.16	208.14	21.72
Soft time windows	10-3-7e	183.31	200.77	22.11
	10-3-7f	206.20	201.05	21.81
	10-7-3g	181.60	190.11	22.96
	10-7-3h	193.17	192.10	22.05
	10-7-3i	210.96	201.52	21.30
	20-10-10a	3600	3600	33.17
	20-10-10b	3600	3600	32.85

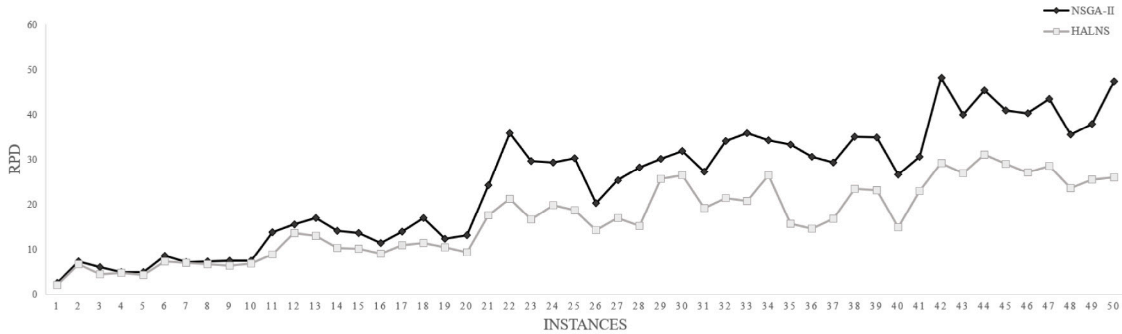
Table 9
The efficiency and effectiveness of the proposed algorithm compared with Li et al. (2016).

Instance	Profit value (€)		CPU time (s)	
	HALNS	Li et al. (2016)	HALNS	Li et al. (2016)
30-15-15c	9546.13	9356.80	42.07	35.70
30-10-20d	11,747.64	11,596.60	56.19	56.30
30-10-20e	10,803.15	10,763.20	44.72	46
30-10-20f	7661.06	7478	59.10	60.50
30-20-10g	11,635.27	10,056.20	41.10	40.07
30-20-10h	11,055.28	10,268.20	46.03	47.80
30-20-10i	9039.75	8278.90	42.93	42
100-50-50a	77,745.22	74,431.90	727.76	741.23
100-50-50b	86,093.82	85,631.40	741.33	766.84
100-50-50c	114,216.05	111,717.10	517.61	515.28
100-25-75d	90,007.44	86,041.30	1007.52	1023
100-25-75e	98,828.63	96,327	914.66	985.18
100-25-75f	84,041.70	82,667.60	611.11	602
100-75-25g	58,759.19	68,543.20	834.71	866.90

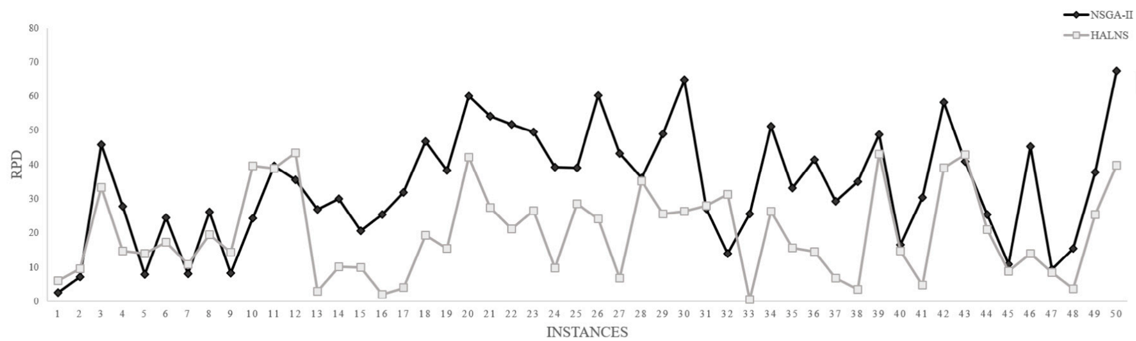
4.4. HALNS vs. NSGA-II

Efficiency and effectiveness are two main criteria for evaluating the performance of an algorithm. In single-objective problems, the value of objective function and computation time, respectively, indicate the effectiveness and efficiency of the algorithm. In bi-objective problems, convergence and diversity of Pareto solutions are two important factors to evaluate them (Deb, 2011). *Time* can still be used as a suitable metric to assess the algorithm's efficiency. However, several metrics have been introduced in the literature for effectiveness measurement. We use *spacing* to quantify the standard deviation of the solution distances in the Pareto front (Rahmati et al., 2013). Moreover, we apply the MID metric used by Rahmati et al. (2013) to measure the accessibility of solutions from an ideal point that is $(0, 0)$ on the Pareto front.

To assess the quality of the proposed algorithm, in this section, we compare its performance with the non-dominated sorting genetic algorithm-II or NSGA-II applied in Mamaghani and Davari (2020) to solve the bi-objective periodic location-routing problem. We chose this algorithm as a benchmark because of its efficiency in solving problems based on periodic VRP. We solve each instance three times to compare the two algorithms. We then calculate the RPD for these six solutions as explained in Section 4.1.2. We then take the average of the three RPD values obtained from each algorithm. Figs. 2–5 depict the performance of each algorithm across various metrics.



(a)
Hard time windows



(b)
Soft time windows

Fig. 2. The Diversity metric for the HALNS and NSGA-II algorithms with hard and soft time windows.

Regarding diversity, as illustrated in Fig. 2(a), it was observed that HALNS generally demonstrates better performance in hard time windows with lower diversity (in RPD, lower values indicate better performance) in all test problems with hard time windows. However, when considering soft time windows, as shown in Fig. 2(b) for 8 instances (1, 2, 5, 7, 9, 10, 12, 43), the performance of NSGA-II exceeds HALNS in this metric. In particular, instances 11, 28, 40, and 47 exhibit identical diversity performance for both algorithms. Fig. 3(a) demonstrates that, on average, HALNS outperforms NSGA-II in terms of MID in all instances, with the difference being insignificant for large-sized instances, particularly within instances 38–50. Switching focus to soft time windows (as depicted in Fig. 3(b)), the proposed algorithm exhibits superior performance in all instances except instances 4 and 13, where NSGA-II surpasses HALNS. Regarding spacing (Fig. 4), we note that HALNS generally exhibits superior performance, displaying lower spacing in most instances. Regarding the time metric, in hard time windows (Fig. 5(a)), the RPD of the two algorithms aligns closely, with an insignificant difference, meaning there is no clear priority between them. However, in soft time windows (Fig. 5(b)), the proposed algorithm generally outperforms NSGA-II in most medium- and small-sized instances. As for large-sized instances, conclusive remarks regarding the running time of the algorithms cannot be made.

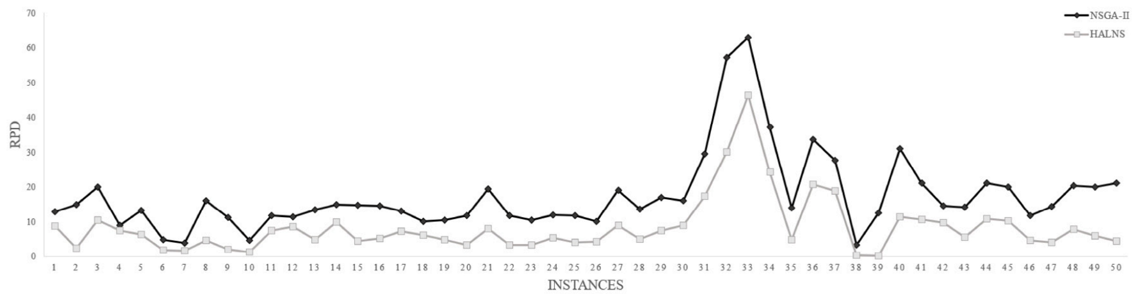
4.5. HALNS vs. MOMA

Multi-objective memetic algorithm (MOMA) is an advanced optimization technique that combines an evolutionary algorithm with local search methods to address problems involving multiple conflicting objectives. By integrating global exploration with local refinement, MOMA can effectively approximate the Pareto front, offering diverse, high-quality solutions . Given these characteristics, MOMA was selected as a comparative benchmark in this study. Its hybrid structure, which aligns conceptually with the HALNS framework, makes it particularly suitable for evaluating the proposed approach under multi-period, bi-objective scenarios. Specifically, MOMA's demonstrated effectiveness in multi-objective combinatorial problems, including vehicle routing and scheduling, allows for a meaningful assessment of convergence behavior, solution diversity, and trade-off quality when compared to HALNS.

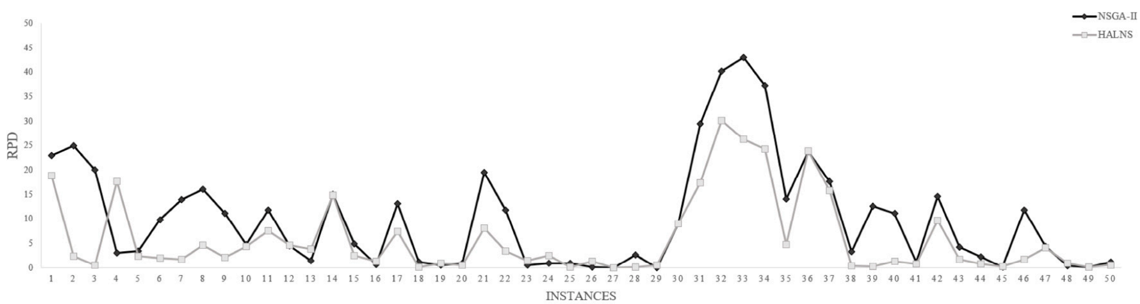
In the context of multi-objective optimization for the green multi-period request assignment problem, a comparative analysis is conducted between the two algorithms. The differences are measured along the four performance indicators for two different settings; hard and soft time windows. The results are illustrated in Figs. 6–9. The objective is to assess the efficiency of MOMA in addressing the same optimization challenges and to determine its suitability relative to HALNS.

The diversity metric, which evaluates the extent of solution spread along the Pareto frontier, reveals notable differences between HALNS and MOMA. HALNS previously demonstrated consistent performance with high diversity under hard time window settings, indicating a well-distributed set of non-dominated solutions. Analysis of the MOMA results under similar constraints shows greater variability and less consistency in solution distribution. The observed clustering of solutions in several instances suggests that while MOMA explores a wide range of the solution space, its coverage lacks the uniformity necessary for robust Pareto approximation. This trend becomes even more evident under soft time window scenarios, where MOMA demonstrates inconsistent distribution patterns, potentially reducing its effectiveness in supporting well-balanced decision-making among trade-off solutions.

Regarding the proximity to the ideal solution, the MID metric underscores HALNS's superior convergence characteristics. The values extracted from MOMA's performance under hard and soft time windows indicate generally higher MID scores, implying that its solutions tend to deviate more significantly from the ideal reference point. This outcome

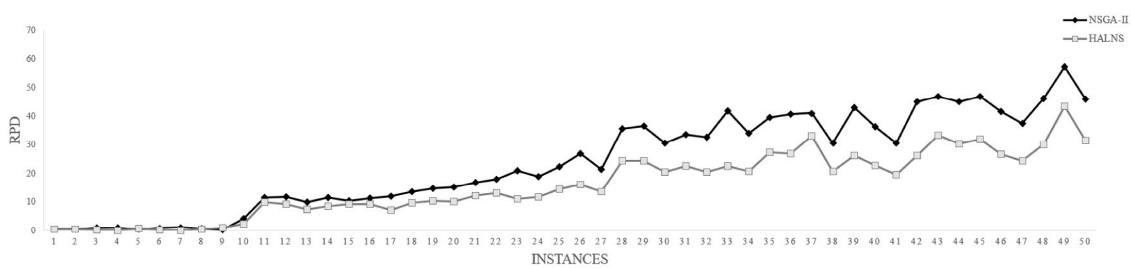


(a)
Hard time windows

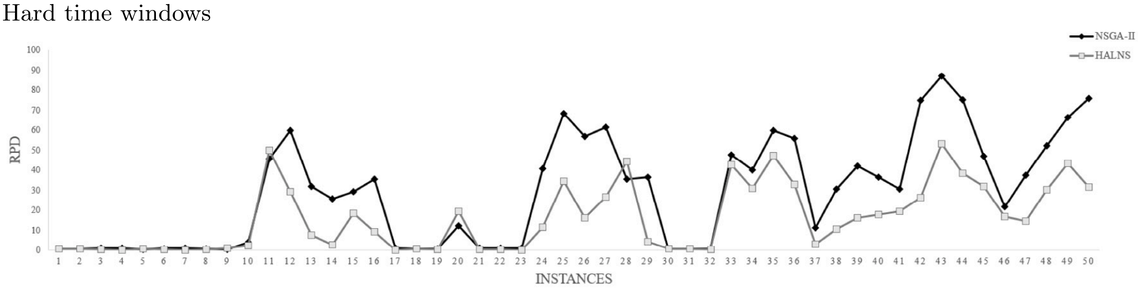


(b)
Soft time windows

Fig. 3. The MID metric for the HALNS and NSGA-II algorithms with hard and soft time windows.



(a)
Hard time windows

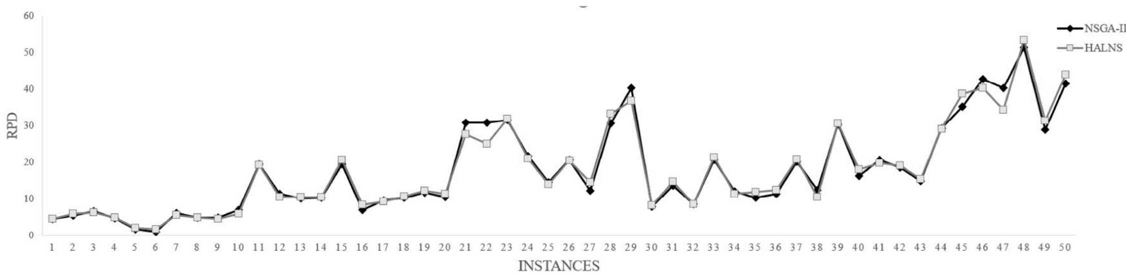


(b)
Soft time windows

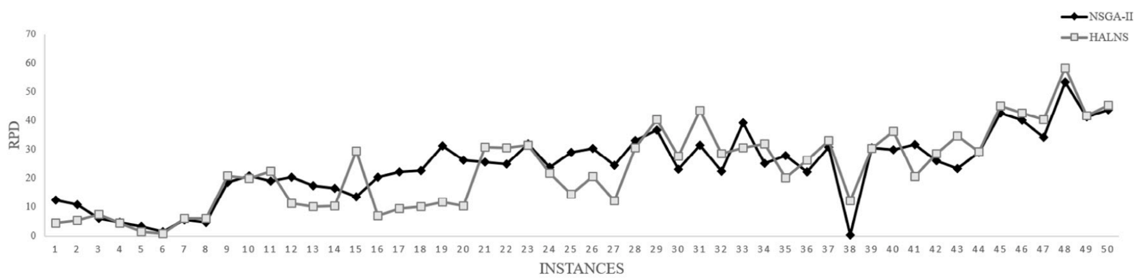
Fig. 4. The Spacing metric for the HALNS and NSGA-II algorithms with hard and soft time windows.

suggests a reduced capability of MOMA in consistently identifying solutions that are simultaneously optimal across all objectives. HALNS, in contrast, generates solutions that not only effectively span the solution space but also remain closely aligned with the theoretical optimum. The spacing metric, which measures the uniformity of solution intervals on the Pareto front, further supports the comparative advantage of HALNS. Lower spacing values are associated with HALNS, indicating a more evenly distributed set of solutions and enhancing the algorithm's practicality in real-world applications where consistent trade-offs are critical. In contrast, MOMA's higher and more dispersed spacing values point to inconsistencies in solution spacing, which may hinder the comprehensive evaluation of the trade-offs between cost and environmental objectives.

Finally, computational efficiency remains a significant criterion for algorithm selection, especially in large-scale, real-time decision making environments. The data on execution time indicates that while MOMA is computationally competitive, HALNS maintains a performance advantage, particularly in instances with increased complexity and scale.

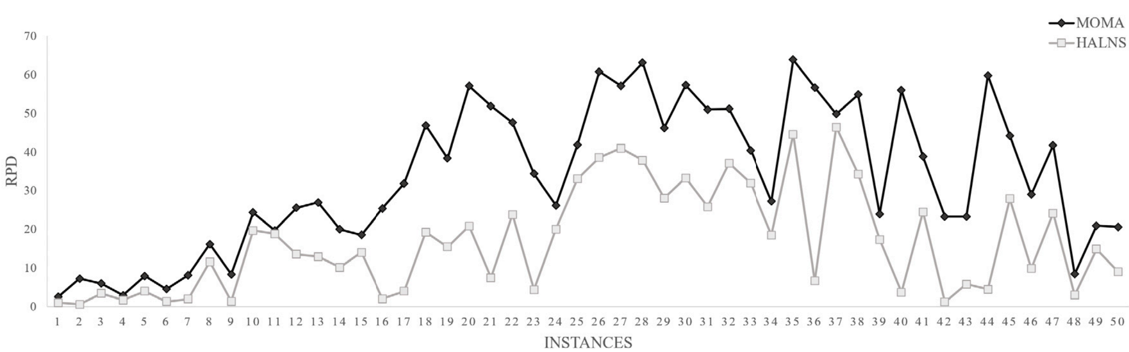


(a)
Hard time windows

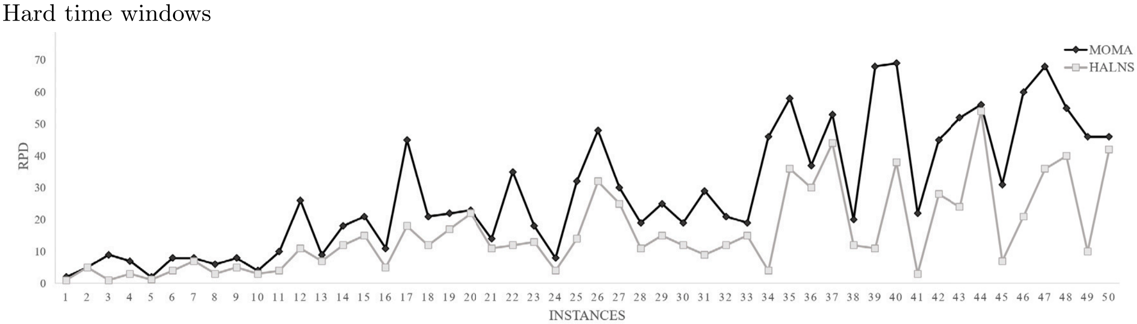


(b)
Soft time windows

Fig. 5. The Time metric for the HALNS and NSGA-II algorithms with hard and soft time windows.



(a)
Hard time windows



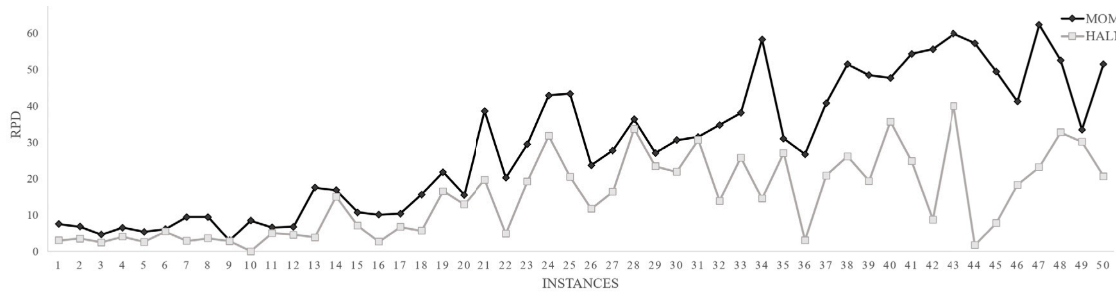
(b)
Soft time windows

Fig. 6. The Diversity metric for the HALNS and MOMA algorithms with hard and soft time windows.

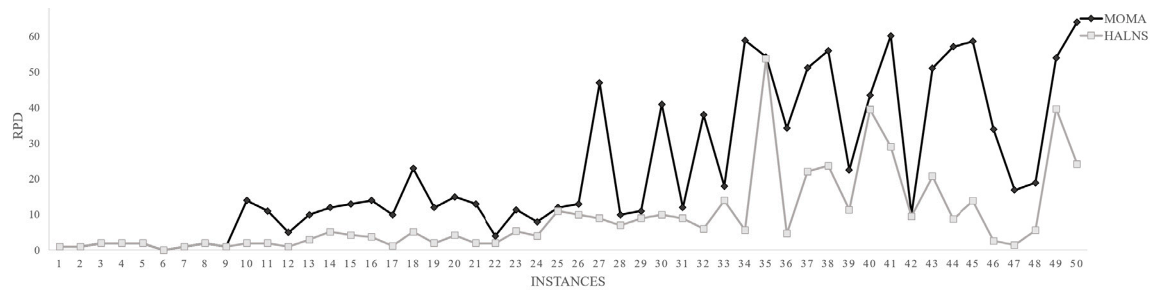
The relatively higher execution times observed in MOMA are likely attributable to its use of intensive local search and genetic operations, which, although potentially beneficial for solution quality, do not consistently translate into superior performance across the other key metrics.

4.6. *HALNS vs. MOSA*

In this section, we perform a comparison between the proposed HALNS algorithm and multi-objective simulated annealing (MOSA) to evaluate their effectiveness in solving the GMP-RAP. Similar to the

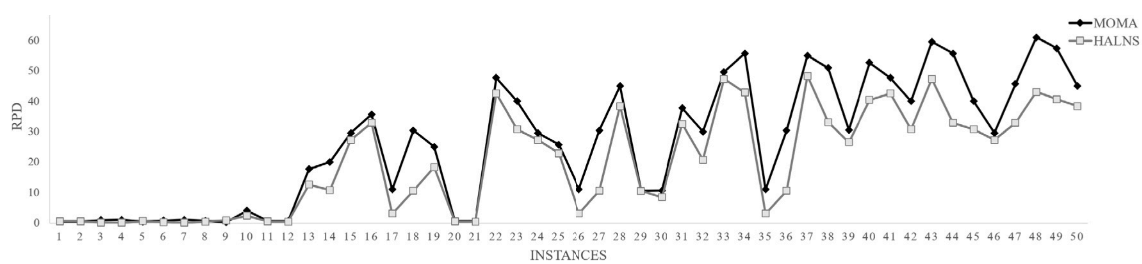


(a)
Hard time windows

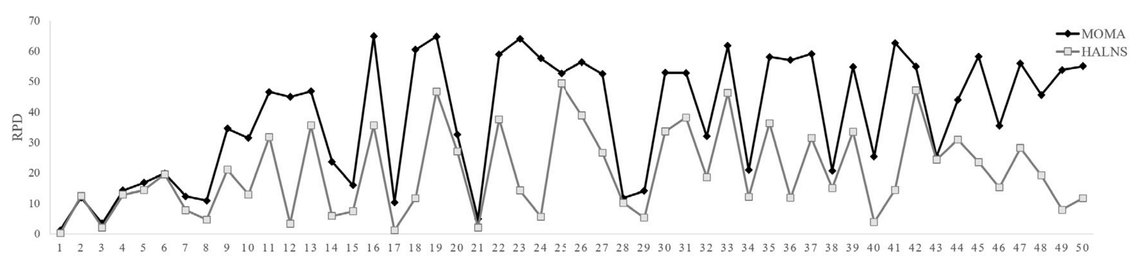


(b)
Soft time windows

Fig. 7. The MID metric for the HALNS and MOMA algorithms with hard and soft time windows.



(a)
Hard time windows

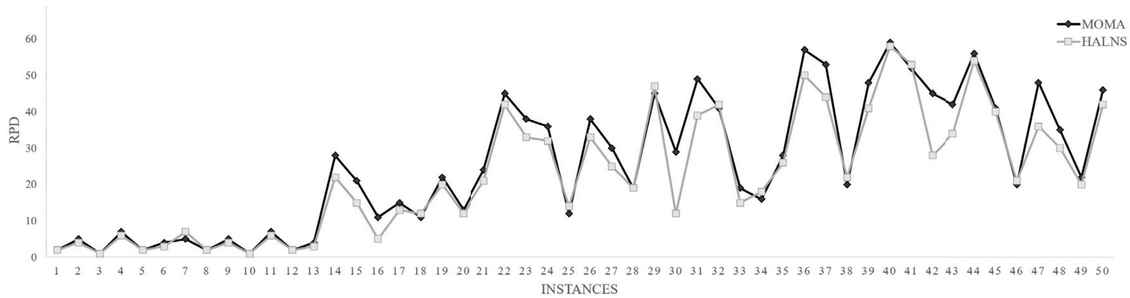


(b)
Soft time windows

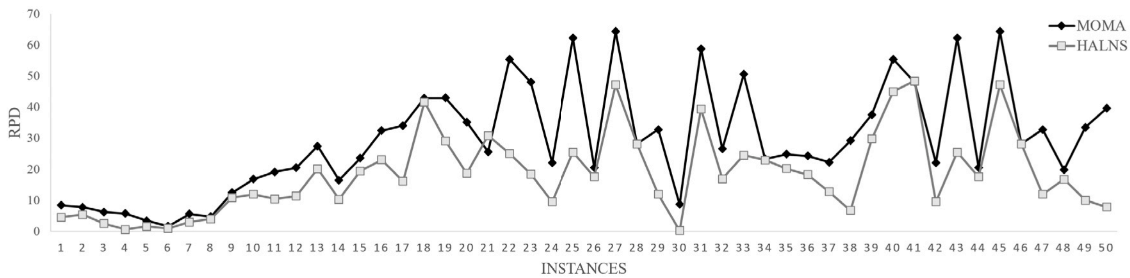
Fig. 8. The Spacing metric for the HALNS and MOMA algorithms with hard and soft time windows.

previous sections, the assessment focuses on convergence, diversity, spacing, and computational time under both hard and soft time window settings. In this study, MOSA was implemented using a weighting approach to scalarize the bi-objective function, enabling it to explore trade-offs within a single-objective simulated annealing framework. MOSA maintains a set of non-dominated solutions and employs a probabilistic acceptance criterion that supports exploration and prevents premature convergence. Due to its simplicity, adaptability, and established success in routing problems, MOSA serves as a relevant and robust baseline for evaluating the performance of HALNS.

Figs. 10–13 illustrate the comparative diversity, MID, spacing, and computational time performance between HALNS and MOSA in scenarios of hard and soft time windows. The outcomes of the comparative analysis reveal several important insights. HALNS consistently achieves better diversity than MOSA under hard and soft time window constraints. The average RPD value in the diversity metric for HALNS



(a)
Hard time windows



(b)
Soft time windows

Fig. 9. The Time metric for the HALNS and MOMA algorithms with hard and soft time windows.

was considerably lower than that of MOSA, indicating that HALNS-generated Pareto solutions were more evenly spread and offered better trade-offs for decision-makers. As shown in Fig. 10, HALNS maintains a more uniform and comprehensive spread across the Pareto front in both cases.

In terms of convergence, HALNS significantly outperforms MOSA across hard and soft time window settings. The average RPD values in the MID metric obtained by HALNS were substantially smaller than those achieved by MOSA, as depicted in Fig. 11. This suggests that HALNS provides a broader spread of solutions and generates solutions much closer to the ideal point, which is essential in multi-objective optimization.

In addition to diversity and convergence, the spacing metric further validates the quality of solutions obtained by HALNS. In both hard and soft time window settings, HALNS achieves significantly lower RPD values compared to MOSA. This indicates that HALNS not only covers a broader range of trade-offs but also distributes solutions more uniformly across the Pareto front. Low RPD values in the spacing metric are particularly important in multi-objective optimization, as they ensure that decision-makers are offered a well-balanced set of alternatives without large gaps. As confirmed by the results, HALNS provides consistently uniform and stable Pareto fronts, whereas MOSA-generated solutions exhibit higher variability in spacing, leading to less reliable distributions.

Investigating the differences between the two methods with respect to execution time shows that HALNS dominates MOSA in almost all instances. This is crucial particularly for large-scale instances. These analyses show that the computational time, as a performance indicator of HALNS is not only acceptable but also should be considered efficient relative to the quality of solutions achieved.

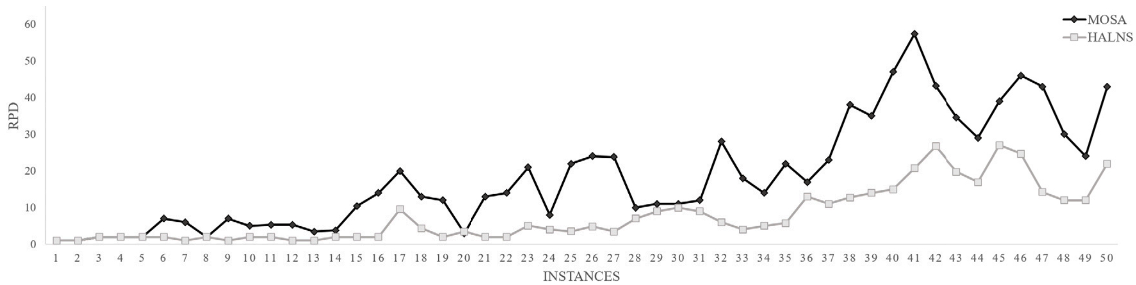
The remarkable performance of HALNS can be attributed to an important factor. HALNS integrates an Adaptive Large Neighborhood Search algorithm with a population-based Tabu Search and a mutation operator, which together enhance both intensification and diversification throughout the search process. In contrast, MOSA predominantly relies on restarting local searches, which increases diversification but lacks a systematic intensification mechanism to exploit promising areas effectively.

4.7. HALNS vs. MOEA

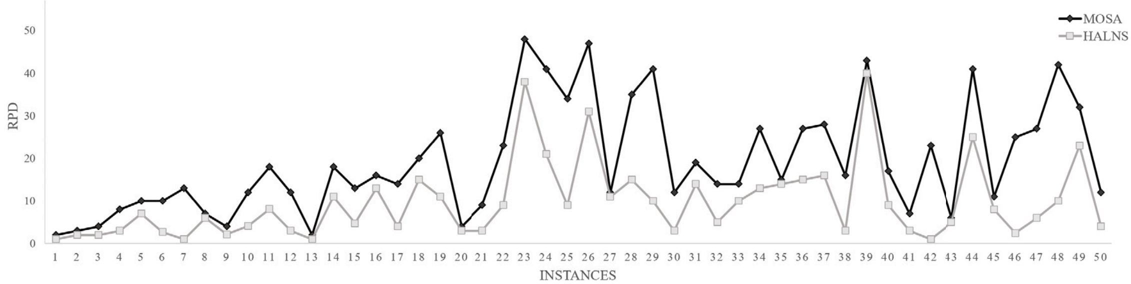
This section compares the performance of the HALNS algorithm with that of MOEA to further validate the robustness and superiority of the proposed method. MOEAs are widely used in multi-objective optimization due to their ability to simultaneously approximate the Pareto front and maintain population diversity. In this study, the MOEA is implemented based on a weighting approach to scalarize the bi-objective function, enabling the algorithm to evaluate trade-offs between cost and environmental impact. Despite their simplicity, MOEAs often suffer from slow convergence and limited local search capabilities, particularly in large and complex routing problems. However, their population-based structure and broad applicability make MOEAs a relevant baseline for assessing solution diversity and convergence behavior in multi-objective vehicle routing settings.

The performance of the algorithms is evaluated based on four criteria: diversity, convergence (measured by MID), spacing, and computational time, under both hard and soft time window constraints. The results are presented in Figs. 14 to 17. As illustrated in the diversity comparison (Fig. 14), HALNS consistently outperforms MOEA across all problem instances under both hard and soft time window conditions. HALNS achieves lower RPD values, reflecting a more uniformly distributed Pareto front. In contrast, MOEA generally exhibits limited diversity, particularly as the instance size increases. HALNS continues to provide broader and more stable coverage of the Pareto front across varying scenarios.

Fig. 15 illustrates the convergence behavior of the algorithms using the MID metric. HALNS consistently outperforms MOEA across all instances, achieving lower RPD values that indicate solutions are closer to the ideal reference point. The convergence performance of MOEA deteriorates notably in medium-sized instances under soft time window constraints and in large-sized instances under hard time window constraints. This consistent advantage emphasizes the effectiveness of HALNS's search intensification mechanisms, attributed to the integration of Adaptive Large Neighborhood Search (ALNS) with a population-based Tabu Search strategy.

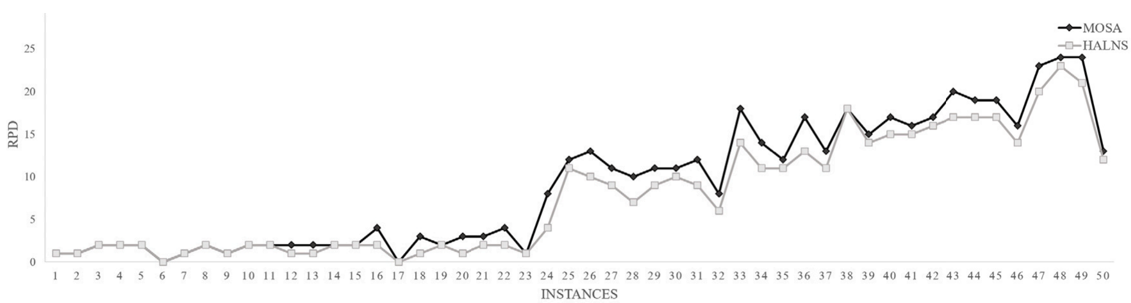


(a)
Hard time windows

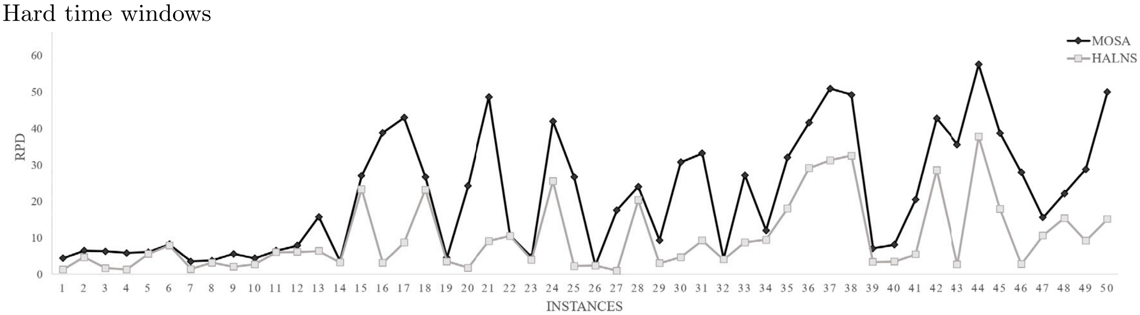


(b)
Soft time windows

Fig. 10. The Diversity metric for the HALNS and MOSA algorithms with hard and soft time windows.



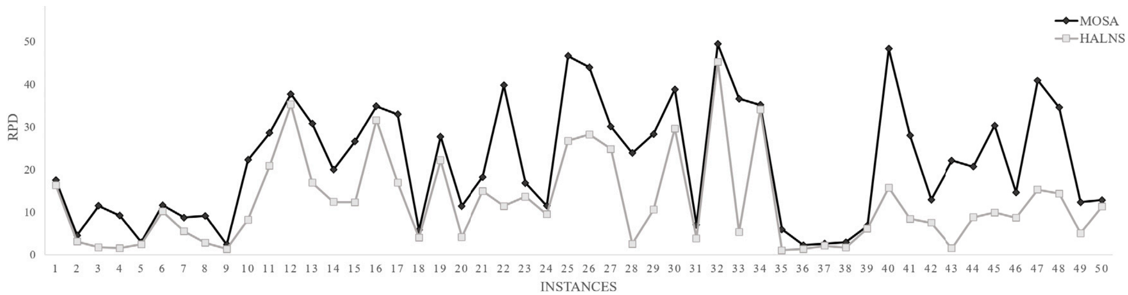
(a)
Hard time windows



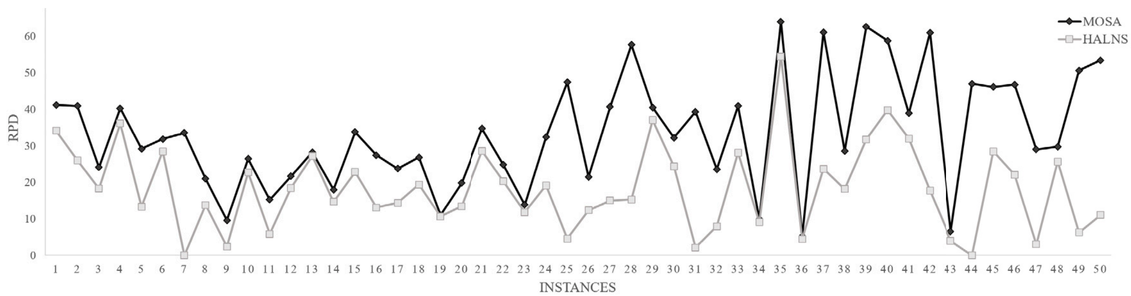
(b)
Soft time windows

Fig. 11. The MID metric for the HALNS and MOSA algorithms with hard and soft time windows.

HALNS demonstrates a distinct performance advantage in both solution quality and computational efficiency. As illustrated in Fig. 16, HALNS achieves significantly lower spacing RPD values, highlighting its superior ability to generate uniformly distributed solutions across the Pareto front. This reflects the algorithm's effective balance between exploration and exploitation. In terms of computational time (Fig. 17), HALNS consistently outperforms MOEA in all instances under soft time and hard time windows constraints. Even in larger instances (both hard and soft time windows), HALNS maintains better efficiency. These results highlight the importance of the strength of HALNS in delivering high-quality, well-distributed solutions with lower computational overhead, making it a robust and scalable approach for solving complex multi-objective problems.

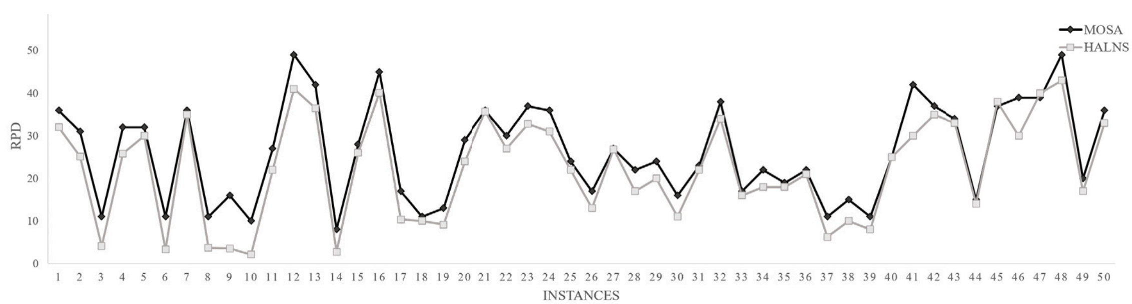


(a)
Hard time windows

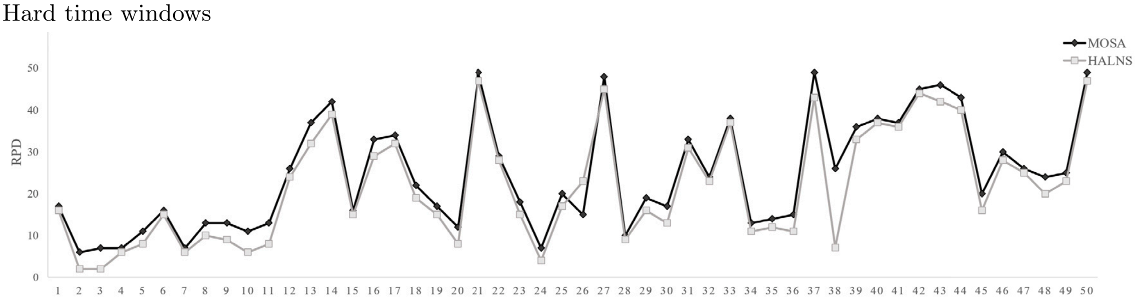


(b)
Soft time windows

Fig. 12. The Spacing metric for the HALNS and MOSA algorithms with hard and soft time windows.



(a)
Hard time windows

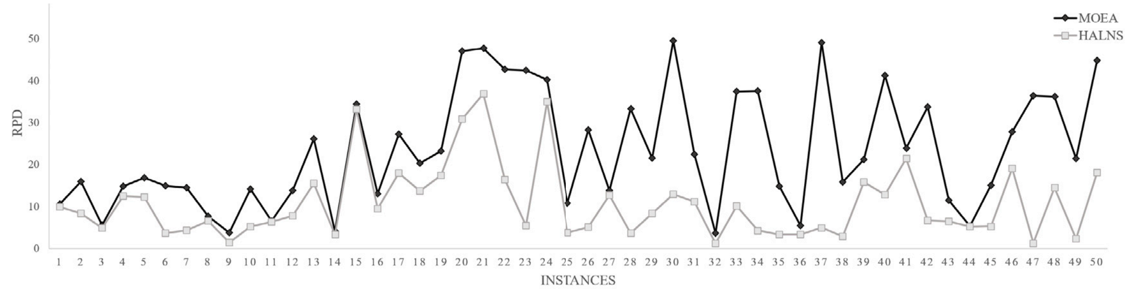


(b)
Soft time windows

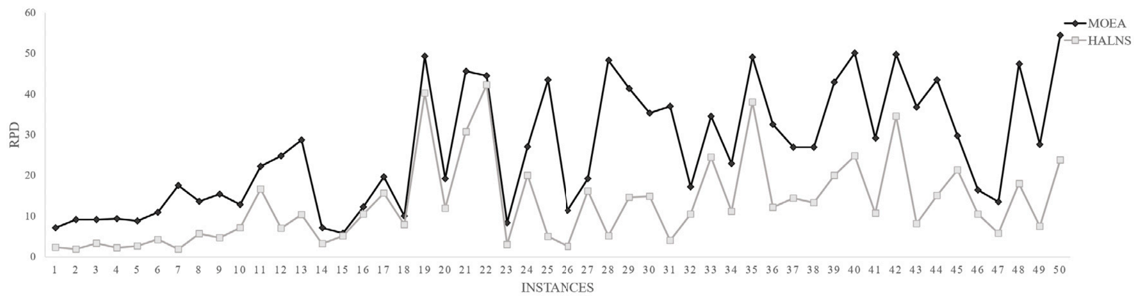
Fig. 13. The Time metric for the HALNS and MOSA algorithms with hard and soft time windows.

5. Managerial insights

We have performed a detailed analysis on the problem presented in the paper to derive practical insights for managers. To achieve this, we have examined various components of the problem for which managers can make better decisions. Moreover, we have quantified and analyzed the impact of such decisions on the results of the system. Specifically, our analysis focuses on three key elements, namely emission targets, fleet size, and time windows, all of which have significant implications for decision-making in practice. Our numerical illustration, supported by a real-world-inspired case study (Appendix B), aims to provide additional insights into optimal policies. In this analysis, we compare

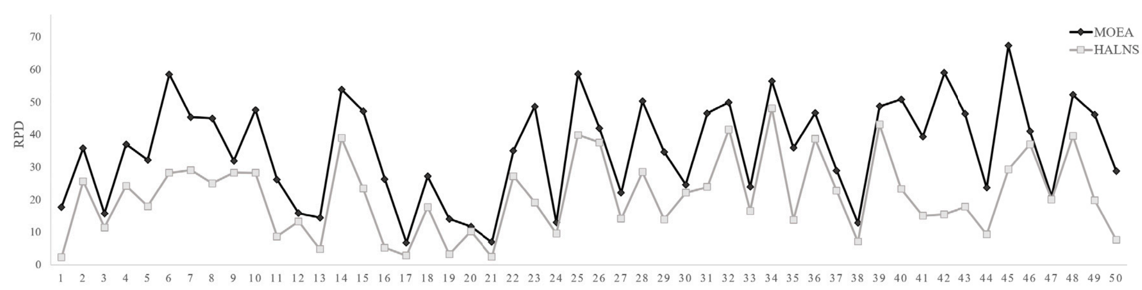


(a)
Hard time windows

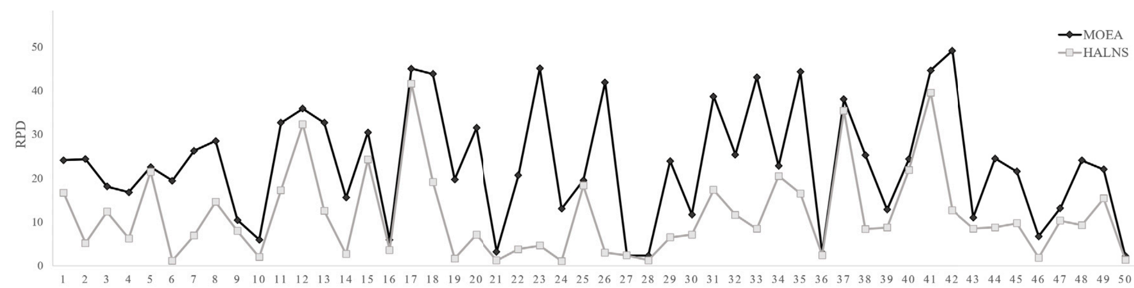


(b)
Soft time windows

Fig. 14. The Diversity metric for the HALNS and MOEA algorithms with hard and soft time windows.

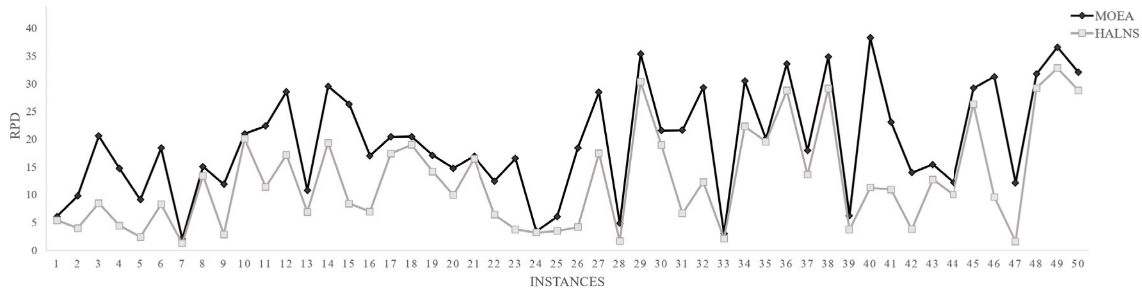


(a)
Hard time windows

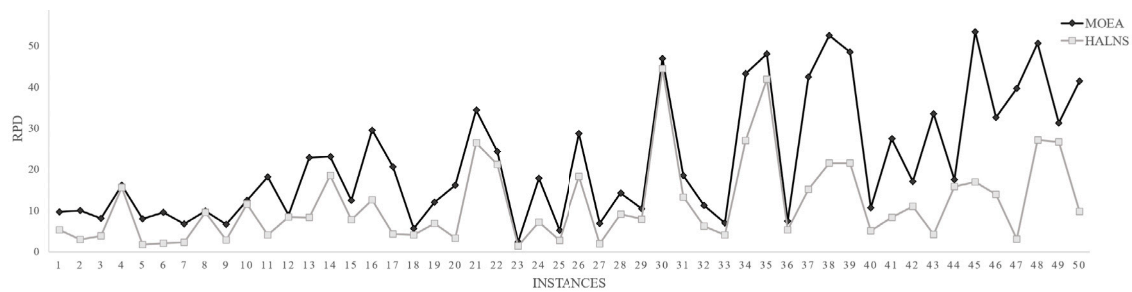


(b)
Soft time windows

Fig. 15. The MID metric for the HALNS and MOEA algorithms with hard and soft time windows.

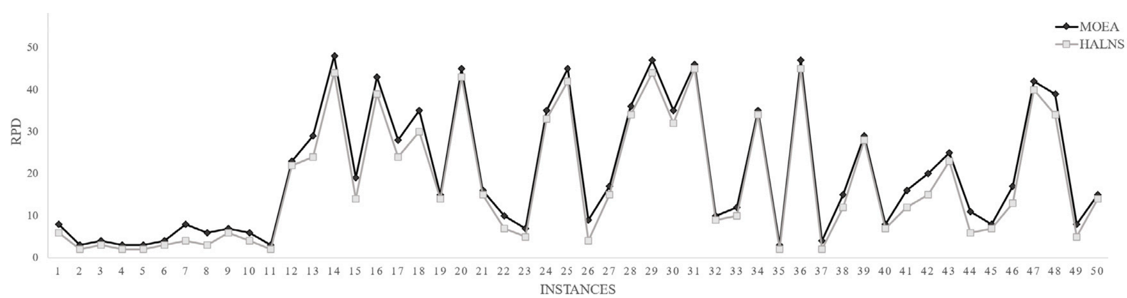


(a)
Hard time windows

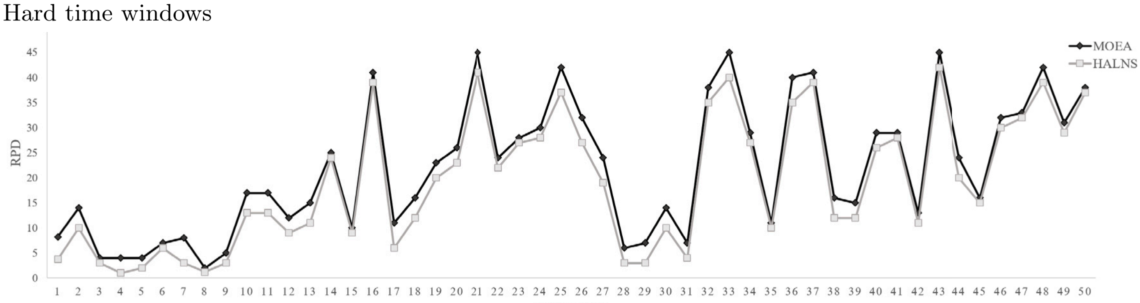


(b)
Soft time windows

Fig. 16. The Spacing metric for the HALNS and MOEA algorithms with hard and soft time windows.



(a)
Hard time windows



(b)
Soft time windows

Fig. 17. The Time metric for the HALNS and MOEA algorithms with hard and soft time windows.

the obtained profits resulting from various strategies. Additionally, we conduct a sensitivity analysis to examine how optimal profits change under different market conditions, particularly concerning parameters.

5.1. Emissions targets

In this section, we study the effect of emission targets on logistics operations and allocation of requests. We assume that the carrier intends to decrease the emissions produced by a certain level. These levels can be determined based on the carrier's initiatives or following the targets set by authorities. To conduct this experiment, we consider two instances, 30-15-15c and 100-50-50b, and solve them initially without considering any targets for CO_2e emissions. This means that we focus on maximizing profit while minimizing emissions. In the next step, we consider the minimum value obtained for emissions in each instance and use this value to set emission targets, i.e., we reduce the

Table 10
The impacts of imposing emissions targets on the problem's solution.

Instance	Reduction (%)	CO_2e emissions (gr)	Profit (€)	Utilization rate (%)	Selective requests served (#)	Vehicles used (#)
30-15-15c with Hard time windows	–	17,813.07	8989.48	38.26	11	9
	10	16,031.76	8704.60	40.05	10	8
	15	15,141.11	8477.33	42.18	9	7
	20	14,250.46	8192.15	45.31	9	7
	25	13,359.80	7831.20	46.37	8	6
	30	12,469.15	7557.04	49.10	7	5
	35	11,578.49	7306.68	52.72	6	4
	40	10,687.84	7067.13	55.44	5	3
30-15-15c with Soft time windows	–	17,107.12	9212.17	41.11	11	9
	10	15,396.41	8923.85	42.56	10	8
	15	14,541.05	8757.19	44.07	9	7
	20	13,685.69	8431.30	47.38	8	6
	25	12,830.34	8161.47	49.15	7	5
	30	11,974.98	7882.28	53.93	6	4
	35	11,119.63	7506.38	55.67	5	3
	40	10,264.27	7383.70	59.48	4	3
100-50-50b with Hard time windows	–	56,046.48	71,158.20	38.15	43	30
	10	50,441.40	66,413.74	39.47	40	29
	15	47,639.51	61,745.46	41.33	39	28
	20	44,837.18	55,816.18	43.10	37	28
	25	42,034.86	48,443.68	45.19	35	27
	30	39,232.53	42,175.06	49.57	32	26
	35	36,430.21	35,971.12	53.09	30	25
	40	33,627.88	30,775.96	57.73	27	24
100-50-50b with Soft time windows	–	55,171.22	75,309.28	40.18	42	30
	10	49,654.09	64,434.93	41.56	41	29
	15	46,895.53	60,963.30	43.80	39	28
	20	44,136.97	55,746.28	44.26	38	28
	25	41,378.41	51,788.91	46.68	36	27
	30	38,619.85	46,440.70	48.12	35	26
	35	35,861.30	43,638.63	51.03	33	25
	40	33,102.73	39,131.55	55.71	31	24

value by 10%, 15%, 20%, 25%, 30%, 35%, and 40%. Finally, we run a profit maximization problem while imposing the emission targets. The results of this experiment are presented in Table 10.

As illustrated in the table, setting emission targets significantly impacts the profits earned. More strict emission targets force the model to utilize the vehicles' capacity as much as possible. This setting also increases the utilization rate and decreases the number of vehicles used. As such, selective requests that are further away from others become less interesting from an environmental perspective, given the high amount of CO_2e emissions produced due to serving them, even if they offer relatively high margins. In such cases, the model seeks clusters of requests that are closely located to each other to ensure a high utilization rate while minimizing marginal costs.

5.2. Fleet size

One of the most critical decisions managers must make in the transportation sector is to determine the appropriate size of their fleet. Given the financial significance of such decisions, it is crucial to understand how they impact the system's performance before taking any action. To perform this analysis, we consider one small and one large instance and increase their fleet sizes by increments of one and three, respectively. This analysis highlights the effects of such investments on a carrier's performance.

In this problem, selective requests could be delivered within a specific period window, and indeed, delivering early in that period window would create higher value for the customer. We measure the improvement of the system in terms of the delivery time of those requests by introducing a new metric, *delivery performance*. This metric can take values between 0% and 100%, where 0% means that all selective requests in the corresponding instance are served during the last period of their period window. In contrast, a value of 100% indicates that all selective requests are received during the first period of their period window. The analysis results are summarized in Table 11.

Intuitively, increasing capacity enables the carriers to serve more selective requests, thereby increasing profits. However, it is important to note the significance of this increase. This experiment also demonstrates that expanding the fleet size leads to improved performance in terms of delivery time, meaning more customers receive their cargo earlier within their period window, especially in large-sized instances. However, this improvement comes with the drawback of elevated emissions. Therefore, managers should conduct a detailed analysis before making such decisions.

5.3. Time windows

One of the restrictive properties of some VRPs is the time windows defined by customers. In the problem discussed in this research, we assume that customers can set two types of time windows (hard and soft) for their shipments. In this section, we explore the impacts of these constraints on the solution. To end this, we investigate two scenarios. First, we examine the original constraints, which include both hard and soft time window constraints, and then solve the problem again without considering the time window constraints. By comparing the results from these two scenarios, we quantify the impact of time windows on operations. This numerical comparison provides valuable insights for managers, helping them understand the implications of time windows on their operations. Based on this analysis, managers can offer customers incentives to relax time windows when feasible, thereby reducing costs and emissions. The results of this analysis are presented in Table 12.

Relaxing time windows significantly enhances the performance of the transportation plan, as evidenced in Table 12. The improvements are found to be substantial. As indicated in the table, when time windows were removed for medium-sized instances, profits increased by 28.74% for problems with soft time window constraints and by 30.46% for problems with hard time window constraints. In large

Table 11
The impacts of fleet size on the problem's solution.

Instance	Vehicles used (#)	Profit (€)	CO_2e emissions (gr)	Utilization rate (%)	Selective requests served (#)	Delivery performance (%)
	9	8989.48	17,813.07	38.26	11	43.26
30-15-15c	10	9850.37	18,721.30	37.20	12	43.87
Hard time windows	11	10,760.22	19,861.47	36.77	13	44.61
	12	10,111.90	21,090.70	36.12	13	45.05
	13	11,740.71	23,751.33	35.35	14	50.17
	9	9212.17	17,107.12	41.11	11	48.17
30-15-15c	10	9075.31	17,963.12	40.87	11	48.86
Soft time windows	11	9765.15	18,455.44	39.26	12	50.24
	12	10,438.75	19,060.13	38.61	13	51.63
	13	12,217.41	21,187.59	36.91	14	55.10
	30	71,158.20	56,046.48	38.15	43	49.92
100-50-50b	33	74,869.16	60,205.71	37.47	45	53.54
Hard time windows	36	76,020.70	63,384.15	36.29	46	55.08
	39	81,390.17	67,578.63	35.14	48	58.61
	42	83,764.25	71,192.58	34.74	49	62.22
	30	75,309.28	55,171.22	40.18	42	52.47
100-50-50b	33	75,960.18	59,620.11	39.37	43	54.60
Soft time windows	36	78,547.33	64,719.90	37.94	45	56.17
	39	83,011.56	67,118.74	35.46	47	59.33
	42	95,763.66	70,016.42	33.80	49	64.48

Table 12
Hard versus soft time windows.

Instance	Time windows	Profit (€)	CO_2e emissions (gr)	Selective requests served (#)	Delivery performance (%)
30-15-15c	Hard	8989.48	17,813.07	11	43.26
	Soft	9212.17	17,107.12	11	48.17
	–	12,928.21	15,377.56	14	55.33
Improvement rate (Hard)		+30.46%	−13.67%		
Improvement rate (Soft)		+28.74%	−10.11%		
100-50-50b	Hard	71,158.20	56,046.48	43	49.92
	Soft	75,309.28	55,171.22	42	52.47
	–	90,130.16	43,380.09	45	60.13
Improvement rate (Hard)		+21.4%	−22.60%		
Improvement rate (Soft)		+16.44%	−21.37%		

instances, profit improvements were 16.44% for soft time windows and 21.04% for hard time windows. Furthermore, executing medium-sized instances without soft time window constraints led to a noteworthy 10.11% reduction in CO_2e emissions. Similarly, for instances without hard time window constraints, there was a significant 13.67% reduction in CO_2e emissions. Furthermore, when running large-sized instances without soft and hard time windows, we observed reductions in CO_2e emissions of 21.37% and 22.60%, respectively.

Beyond the improvements in the objective functions, there are notable delivery times enhancements. This experiment aims to quantitatively assess the impact of time windows on solutions, providing managers with insights to make informed trade-offs. By analyzing the effects of time windows on profit and CO_2e emissions, this study helps managers to weigh the advantages and disadvantages of implementing time windows and make decisions accordingly.

6. Conclusions

We have conducted a comprehensive study addressing a request assignment problem within the context of the pickup and delivery problem, referred to as A green multi-period request assignment problem in road freight transport (GMP-RAP). This problem considers two distinct requests: reserved requests, which are mandatory to serve within predefined periods, and selective requests, which are optional. For selective requests, a specific period window exists during which they can be accommodated, with a preference for serving them as close as possible to their lower bound.

Our formulated problem combines economic and environmental objectives, framed as a bi-objective mixed-integer programming formulation, featuring hard and soft time windows and a fleet of heterogeneous vehicles. The first objective function seeks to maximize total profit, while the second objective aims to minimize CO_2e emissions. To tackle this problem, we employed various solution methods, including exact ϵ-constraint optimization, weighting methods, and, particularly for larger instances due to the problem's NP-hard nature, a novel hybrid Adaptive Large Neighborhood Search (HALNS) algorithm incorporating TS_{pop} (population-based TS with a mutation operator) and four local search strategies. To improve the efficiency of our algorithm, we utilized the Taguchi approach to fine-tune the HALNS's parameters, significantly contributing to its improved performance.

Our extensive computational results were compared with NSGA-II, MOMA, MOSA, and MOEA across four essential metrics: CPU time, diversity, MID, and spacing. The performance of each algorithm was assessed against these metrics, highlighting the efficiency and effectiveness of our proposed hybrid algorithm. Furthermore, we provided managerial insights, focusing on the influence of key parameters such as the number of vehicles, CO_2e emissions, and time windows on the studied bi-objective formulation. This analysis provides valuable guidance for smaller logistics companies by helping them optimize their operations by making informed decisions. Companies can strategically adjust fleet size and time window settings to maximize profits while minimizing CO_2e emissions.

Although this study offers valuable contributions, it is important to acknowledge certain limitations. First, the problem formulation assumes deterministic parameters, such as demand, travel times, and costs. However, in real-world road freight transport, these factors are often subject to variability. Secondly, while our study focuses on economic and environmental objectives, other relevant sustainability factors, such as social considerations (e.g., driver satisfaction, working

Table 13
Problem context overview.

Aspect	Details
Planning horizon	Five days (Monday–Friday), with reserved requests (pre-assigned orders) and selective requests (optional assignments).
Fleet composition	10 heterogeneous vehicles: - 4 electric vans (Capacity: 10 parcels, Cost: €15 per trip, CO_2 emissions: 0 g/km). - 6 diesel trucks (Capacity: 40 parcels, Cost: €25 per trip, CO_2 emissions: 210 g/km).
Pickup and delivery requests	- Time-sensitive e-commerce orders must be picked up from suppliers and delivered to customers within specific time windows. - Flexible bulk shipments can be assigned dynamically to improve vehicle utilization.
Optimization goals	Optimize pickup and delivery assignment, fleet utilization, and routing while maintaining profitability and sustainability.

hours, and equity in workload distribution), remain unexplored. Integrating such aspects could provide a more holistic approach to sustainable logistics planning.

Future research could also explore exact algorithmic approaches, such as column generation, to improve solution quality, particularly for large-scale instances. Furthermore, implementing the proposed problem in a real-world setting with realistic parameters would allow a more comprehensive evaluation of its tangible impacts on profits and CO_2e emissions.

CRediT authorship contribution statement

Elham Jelodari Mamaghani: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Conceptualization. **Yousef Ghiami:** Writing – original draft, Supervision, Methodology, Conceptualization. **Emrah Demir:** Writing – review & editing, Supervision, Methodology, Conceptualization. **Tom Van Woensel:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We would like to express our sincere gratitude to all those who contributed to the completion of this paper. We thank the Editor and the three anonymous reviewers for their valuable feedback and insightful comments, which greatly improved the quality of this paper.

Appendix A. Flowchart

Fig. 18 gives an overview of all the components of the proposed solution method.

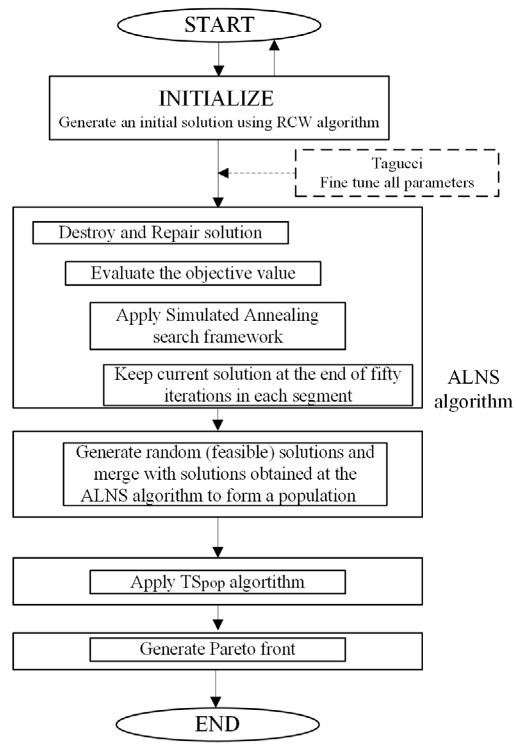


Fig. 18. The flowchart of the solution methodology.

Appendix B. Case study

We present a case study inspired by real-world operations of an urban LTL freight carrier in Amsterdam, managing PD logistics for a retailer handling both perishable and ambient goods. Table 13 provides an overview of the case. Using the HALNS algorithm, we optimize key operational decisions, including request assignment, vehicle selection, PD sequencing, and routing over a five-day planning horizon (Monday–Friday). The carrier operates a heterogeneous fleet with varying capacities, costs, and emissions. Time-sensitive shipments must be picked up and delivered within strict time windows, while ambient goods can be scheduled dynamically based on vehicle capacity and routing efficiency. The objective is to improve vehicle utilization and routing strategies while maintaining cost-effectiveness and minimizing environmental impact.

Table 14 details the assignment of requests: reserved requests, such as perishable goods, follow fixed service days, while selective requests, such as ambient products, are dynamically scheduled based on vehicle capacity and routing efficiency.

Table 14
Assignment of requests to days of the week.

Day	Total requests	Reserved requests	Selective requests
Monday	48	35	13
Tuesday	55	38	17
Wednesday	42	30	12
Thursday	50	36	14
Friday	47	32	15

Vehicles are assigned based on cost, capacity, and emission constraints to optimize efficiency as shown in Table 15. The HALNS algorithm optimizes the order in which requests are served by ensuring that pickup always precedes delivery for all shipments. It effectively clusters nearby requests, minimizing unnecessary travel and improving overall efficiency. For instance, the algorithm consolidates deliveries into a single optimized route rather than making three separate trips to a distribution hub, significantly reducing travel time and cost.

The optimized routing strategy effectively reduces total travel distance and minimizes empty mileage, leading to greater efficiency. For example, a route initially spanned 140 km was optimized to 105 km, resulting in fuel savings and lower emissions (Table 15).

Table 15
Vehicle assignment to requests.

Vehicle type	Assigned deliveries	Avg. distance (km)	CO_2 emissions (kg)
Electric van	18	12	0
Diesel truck	24	50	10.50

Table 16
Optimized routing.

Metric	Before optimization	After optimization	Improvement (%)
Total distance (km)	140	105	25
Empty mileage (km)	40	18	55
CO_2 emissions (kg)	18.20	12.50	31
Fuel cost (€)	120	98	18

The optimized PD schedule brings significant improvements, including a 31% reduction in CO_2e emissions, aligning with sustainability goals (Table 16). Additionally, vehicle utilization improved by 28%, resulting in fewer trips and cost savings, while operational costs decreased by 18%, enhancing profitability without compromising service levels. Furthermore, fleet efficiency increased, enabling the company to handle 15% more requests using the same resources. This case study demonstrates how the proposed approach optimizes PD logistics, serving as a practical decision-making tool for logistics providers and freight carriers.

Data availability

Data will be made available on request.

References

Al Chami, Z., Manier, H., Manier, M.-A., 2019. A lexicographic approach for the bi-objective selective pickup and delivery problem with time windows and paired demands. Ann. Oper. Res. 273, 237–255.

Allahyari, S., Yaghoubi, S., Van Woensel, T., 2021. A novel risk perspective on location-routing planning: An application in cash transportation. Transp. Res. Part E: Logist. Transp. Rev. 150, 102356.

Ben-Said, A., Moukrim, A., Guibadj, R.N., Verny, J., 2022. Using decomposition-based multi-objective algorithm to solve selective pickup and delivery problems with time windows. Comput. Oper. Res. 105867.

Christiaens, J., Vanden Berghe, G., 2020. Slack induction by string removals for vehicle routing problems. Transp. Sci. 54 (2), 417–433.

Deb, K., 2011. Multi-objective optimisation using evolutionary algorithms: An introduction. In: Multi-Objective Evolutionary Optimisation for Product Design and Manufacturing. Springer, pp. 3–34.

Demir, E., Bektaş, T., Laporte, G., 2012. An adaptive large neighborhood search heuristic for the pollution-routing problem. European J. Oper. Res. 223 (2), 346–359.

Demir, E., Bektaş, T., Laporte, G., 2014a. The bi-objective pollution-routing problem. European J. Oper. Res. 232 (3), 464–478.

Demir, E., Bektaş, T., Laporte, G., 2014b. A review of recent research on green road freight transportation. European J. Oper. Res. 237 (3), 775–793.

Demir, E., Syntetos, A., van Woensel, T., 2022. Last mile logistics: Research trends and needs. IMA J. Manag. Math. 33 (4), 549–561.

Du, J., Zhang, Z., Wang, X., Lau, H.C., 2023. A hierarchical optimization approach for dynamic pickup and delivery problem with LIFO constraints. Transp. Res. Part E: Logist. Transp. Rev. 175, 103131.

Evans, G.W., 1984. An overview of techniques for solving multiobjective mathematical programs. Manag. Sci. 30 (11), 1268–1282.

Galiullina, A., Mutlu, N., Kinable, J., Van Woensel, T., 2024. Demand steering in a last-mile delivery problem with home and pickup point delivery options. Transp. Sci..

Ghiami, Y., Demir, E., Van Woensel, T., Christiansen, M., Laporte, G., 2019. A deteriorating inventory routing problem for an inland liquefied natural gas distribution network. Transp. Res. Part B: Methodol. 126, 45–67.

Glover, F., 1989. Tabu search—part I. ORSA J. Comput. 1 (3), 190–206.

Grendreau, M., Potvin, J., 2010. Handbook of Metaheuristics. In: International Series in Operations Research & Management Science, vol. 146, Springer, USA.

Hajipour, V., Farahani, R.Z., Fattahi, P., 2016. Bi-objective vibration damping optimization for congested location–pricing problem. Comput. Oper. Res. 70, 87–100.

Li, Y., Chen, H., Prins, C., 2016. Adaptive large neighborhood search for the pickup and delivery problem with time windows, profits, and reserved requests. European J. Oper. Res. 252 (1), 27–38.

Lyu, Z., Yu, A.J., 2023. The pickup and delivery problem with transshipments: Critical review of two existing models and a new formulation. European J. Oper. Res. 305 (1), 260–270.

Madankumar, S., Rajendran, C., 2018. Mathematical models for green vehicle routing problems with pickup and delivery: A case of semiconductor supply chain. Comput. Oper. Res. 89, 183–192.

Mamaghani, E.J., Davari, S., 2020. The bi-objective periodic closed loop network design problem. Expert Syst. Appl. 144, 113068.

McKinnon, A., Browne, M., Whiteing, A., Piecyk, M., 2015. Green Logistics: Improving the Environmental Sustainability of Logistics. Kogan Page Publishers.

Meng, S., Chen, Y., Li, D., 2024. The multi-visit drone-assisted pickup and delivery problem with time windows. European J. Oper. Res. 314 (2), 685–702.

Moghdani, R., Salimifard, K., Demir, E., Benyettou, A., 2021. The green vehicle routing problem: A systematic literature review. J. Clean. Prod. 279, 123691.

Moradi, H., Zandieh, M., Mahdavi, I., 2011. Non-dominated ranked genetic algorithm for a multi-objective mixed-model assembly line sequencing problem. Int. J. Prod. Res. 49 (12), 3479–3499.

Parragh, S.N., Schmid, V., 2013. Hybrid column generation and large neighborhood search for the dial-a-ride problem. Comput. Oper. Res. 40 (1), 490–497.

Prodhon, C., 2011. A hybrid evolutionary algorithm for the periodic location-routing problem. European J. Oper. Res. 210 (2), 204–212.

Rahmati, S.H.A., Zandieh, M., Yazdani, M., 2013. Developing two multi-objective evolutionary algorithms for the multi-objective flexible job shop scheduling problem. Int. J. Adv. Manuf. Technol. 64 (5–8), 915–932.

Ropke, S., Pisinger, D., 2006. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transp. Sci. 40 (4), 455–472.

Sadeghi, J., Sadeghi, S., Niaki, S.T.A., 2014. A hybrid vendor managed inventory and redundancy allocation optimization problem in supply chain management: An NSGA-II with tuned parameters. Comput. Oper. Res. 41, 53–64.

Santiyuda, G., Wardoyo, R., Pulungan, R., Vincent, F.Y., 2024. Multi-objective reinforcement learning for bi-objective time-dependent pickup and delivery problem with late penalties. Eng. Appl. Artif. Intell. 128, 107381.

Sartori, C.S., Buriol, L.S., 2020. A study on the pickup and delivery problem with time windows: Matheuristics and new instances. Comput. Oper. Res. 124, 105065.

Savelsbergh, M.W., Sol, M., 1995. The general pickup and delivery problem. Transp. Sci. 29 (1), 17–29.

Sels, V., Vanhoucke, M., 2012. A hybrid genetic algorithm for the single machine maximum lateness problem with release times and family setups. Comput. Oper. Res. 39 (10), 2346–2358.

Shaw, P., 1998. Using constraint programming and local search methods to solve vehicle routing problems. In: Principles and Practice of Constraint Programming—CP98: 4th International Conference, CP98 Pisa, Italy, October 26–30, 1998 Proceedings 4. Springer, pp. 417–431.

Soleimani, H., Chaharlang, Y., Ghaderi, H., 2018. Collection and distribution of returned-remanufactured products in a vehicle routing problem with pickup and delivery considering sustainable and green criteria. J. Clean. Prod. 172, 960–970.

Soysal, M., Cimen, M., Demir, E., 2018. On the mathematical modeling of green one-to-one pickup and delivery problem with road segmentation. J. Clean. Prod. 174, 1664–1678.

Sun, P., Veelenturf, L.P., Hewitt, M., Van Woensel, T., 2020. Adaptive large neighborhood search for the time-dependent profitable pickup and delivery problem with time windows. Transp. Res. Part E: Logist. Transp. Rev. 138, 101942.

Sun, W., Yu, Y., Wang, J., 2019. Heterogeneous vehicle pickup and delivery problems: Formulation and exact solution. Transp. Res. Part E: Logist. Transp. Rev. 125, 181–202.

Vaziri, S., Etebari, F., Vahdani, B., 2019. Development and optimization of a horizontal carrier collaboration vehicle routing model with multi-commodity request allocation. J. Clean. Prod. 224, 492–505.

Wang, Z., Dessouky, M., Van Woensel, T., Ioannou, P., 2023b. Pickup and delivery problem with hard time windows considering stochastic and time-dependent travel times. EURO J. Transp. Logist. 12, 100099.

Wang, Y., Peng, S., Zhou, X., Mahmoudi, M., Zhen, L., 2020. Green logistics location-routing problem with eco-packages. Transp. Res. Part E: Logist. Transp. Rev. 143, 102118.

Wang, Y., Wang, X., Wei, Y., Sun, Y., Fan, J., Wang, H., 2023a. Two-echelon multi-depot multi-period location-routing problem with pickup and delivery. Comput. Ind. Eng. 182, 109385.

Wang, Y., Zhang, J., Assogba, K., Liu, Y., Xu, M., Wang, Y., 2018. Collaboration and transportation resource sharing in multiple centers vehicle routing optimization with delivery and pickup. Knowl.-Based Syst. 160, 296–310.

Wolfinger, D., 2021. A large neighborhood search for the pickup and delivery problem with time windows, split loads and transshipments. Comput. Oper. Res. 126, 105110.

Xue, G., 2022. A two-stage heuristic solution for multi-depot collaborative pickup and delivery network with transfers to reduce carbon emissions. J. Clean. Prod. 373, 133839.

Yang, J., Li, Y., 2023. A multicommodity pickup and delivery problem with time windows and handling time in the omni-channel last-mile delivery. Int. Trans. Oper. Res..

Yang, J., Li, Y., 2025. A multicommodity pickup and delivery problem with time windows and handling time in the omni-channel last-mile delivery. Int. Trans. Oper. Res. 32 (3), 1524–1565.

Zarandi, M.F., Mosadegh, H., Fattahi, M., 2013. Two-machine robotic cell scheduling problem with sequence-dependent setup times. Comput. Oper. Res. 40 (5), 1420–1434.

Zhang, K., Li, M., Wang, J., Li, Y., Lin, X., 2023. A Two-stage Learning-based method for Large-scale On-demand pickup and delivery services with soft time windows. Transp. Res. Part C: Emerg. Technol. 151, 104122.

Zhao, J., Dong, H., Wang, N., 2023. Green split multiple-commodity pickup and delivery vehicle routing problem. Comput. Oper. Res. 106318.

Zitzler, E., 1999. Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. Vol. 63, ETH, Switzerland.