The selective multiple depot pickup and delivery problem with multiple time windows and paired demand

Roelink, Daniël; Campuzano, Giovanni; Mes, Martijn; Lalla-Ruiz, Eduardo

DOI
10.1016/j.orp.2025.100342

Publication date
2025

Document Version
Final published version

Published in
Operations Research Perspectives

Operations Research Perspectives 14 (2025) 100342

Contents lists available at ScienceDirect

Operations Research Perspectives

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

The selective multiple depot pickup and delivery problem with multiple time windows and paired demand

Daniël Roelink [a], Giovanni Campuzano [b,c], Martijn Mes [a], Eduardo Lalla-Ruiz [a],*

[a] Department of High-Tech Business and Entrepreneurship, University of Twente, Drienerlolaan 5, Enschede, 7522 NB, The Netherlands
[b] Facultad de Ingeniería, Arquitectura y Diseño, Universidad San Sebastian, Lago Panguipulli 1390, Puerto Montt, 5501842, Chile
[c] Department of Maritime and Transport Technology, Delft University of Technology, Mekelweg 2, Delft, 2628 CD, The Netherlands

ARTICLE INFO

Keywords:
Vehicle Routing Problem
Adaptive Large Neighborhood Search
Simulated Annealing
Backhauling
Freight selection
Freight exchange

ABSTRACT

A recurring challenge for transportation companies is the inefficiency of returning (partially) empty vehicles, or backhauling, after delivering orders. To address this issue, companies search on freight exchange platforms for profitable pickup and delivery orders, aiming to reduce the costs associated with empty return trips. The increasing reliance on freight exchange platforms presents both an opportunity and a challenge: while they offer access to profitable loads, effectively selecting the right combination of orders to maximize returns is challenging. This paper addresses this challenge by introducing the Selective Multiple Depot Pickup and Delivery Problem with Multiple Time Windows and Paired Demand (SMDPDPMTWPD). We formulate the SMDPDPMTWP as a Mixed-Integer Linear Program (MILP) to maximize profit and optimize freight selection for return trips. In addition to the main model, three problem extensions are proposed: (*i*) profit maximization including CO_2 costs, (*ii*) soft time windows, and (*iii*) soft time windows including CO_2 costs. Given the complexity of the problem, we develop an Adaptive Large Neighborhood Search (ALNS) metaheuristic to solve large instances within reasonable computing times and compare it with a Simulated Annealing (SA) heuristic. Results show that ALNS outperforms SA and finds the same optimal solutions as the MILP formulation for small instances. Furthermore, ALNS achieves an average improvement of 308.17% over the initial solutions for the profit maximization variant. The model variant with CO_2 costs shows a slight sensitivity of the routing schedules to the CO_2 emissions costs, whereas we observe a significant change when allowing soft time windows. Finally, soft time windows significantly increase the profits earned compared to the hard time windows (179.54% on average), due to the additional flexibility created when late arrivals are possible.

1. Introduction

Road transport continues to play a crucial role in global logistics. According to the Netherlands Statistics Bureau, the share of road transport in total freight transport in the Netherlands increased from 40.65% in 2016 to 43.26% in 2022 [1]. However, in the same period, both transport prices and the number of empty kilometers have increased, specifically, the latter increased from 25.60% to 26.14% ([2,3]). Because empty kilometers are inefficient, unprofitable and contribute to increasing transportation costs, transportation companies are exerting strong efforts to minimize their occurrence. Besides the economic impact of empty kilometers, there is also an environmental impact in the form of CO_2 pollutants and other emissions [4]. Due to the heightened global awareness of reducing CO_2 emissions and the European Union's adoption of the Corporate Sustainability Reporting Directive (CSRD) in 2021, which mandates that companies should track and mitigate their environmental impact, reducing the number of empty kilometers is a necessity for transportation companies.

Empty kilometers may be inevitable, especially in certain sectors, as evidenced in construction. Still, they can be effectively reduced in other sectors, such as international transport, where empty kilometers typically occur on return trips. Transportation companies try to reduce empty kilometers by enhancing planning efforts and optimizing the routes to be traveled by trucks. However, a transportation company may lack a sufficient or diverse order set to create efficient routes with a low number of empty kilometers. In those cases, transportation companies can utilize so-called freight exchange platforms. These platforms facilitate the exchange of orders among members and connect transportation companies with available capacity to freight providers in need of transportation services. In other words, freight exchange

* Corresponding author.
E-mail addresses: dj.roelink@outlook.com (D. Roelink), giovanni.campuzano@uss.cl (G. Campuzano), m.r.k.mes@utwente.nl (M. Mes), e.a.lalla@utwente.nl (E. Lalla-Ruiz).

https://doi.org/10.1016/j.orp.2025.100342
Received 23 July 2024; Received in revised form 9 March 2025; Accepted 13 May 2025
Available online 31 May 2025

platforms provide a marketplace where transportation companies with excess orders can offer them to other companies facing a shortage of orders, enabling them to bid on these opportunities. However, selecting freights on these platforms is often a manual and time-consuming process that relies on the skills, knowledge, and experience of human planners.

In this paper, we introduce a novel variant of the Vehicle Routing Problem (VRP) called the Selective Multiple Depot Pickup and Delivery Problem with Multiple Time Windows and Paired Demand (SMDPDPMTWPD). This problem models a truck backhauling system in which companies utilize freight exchange platforms to identify and acquire profitable orders selectively, leveraging empty trucks to execute deliveries. The main motivation for studying the SMDPDPMTWPD is to optimize truck backhauling operations through freight exchange platforms, reducing empty miles, lowering emissions, and alleviating road congestion.

As an inherently NP-hard problem, the SMDPDPMTWPD presents significant computational challenges, especially for large-scale instances. To effectively address these complexities, advanced metaheuristic algorithms are essential. As noted in [5], the existing literature on multi-depot backhauling with selective orders is limited, highlighting the importance of this research. This paper addresses the identified research gap (see Section 2.4) by integrating features such as (*i*) truck capacity constraints, (*ii*) selective orders, (*iii*) multiple time windows, and (*iv*) multiple depots, across four distinct objective functions. The main contributions of this paper are as follows.

- We introduce a novel extension of the VRP that addresses the emerging challenges faced by logistics companies using freight exchange platforms. This problem is referred to as the Selective Multiple Depot Pickup and Delivery Problem with Multiple Time Windows and Paired Demand (SMDPDPMTWPD). We formulate the SMDPDPMTWPD as a MILP to optimize the non-compulsory pickup of orders for return trips. Additionally, We extend the SMDPDPMTWPD to proposed three problem variants: (*i*) profit maximization including CO_2 costs, (*ii*) soft time windows, and (*iii*) soft time windows including CO_2 costs.
- We develop an effective Adaptive Large Neighborhood Search (ALNS) metaheuristic with a Simulated Annealing acceptance criterion to solve the SMDPDPMTWPD. To enhance the performance of ALNS, several destroy and repair operators are adapted and tailored from the literature. ALNS can be seamlessly connected to freight exchange platforms to reduce costs and increase revenues for transportation companies.
- To evaluate our optimization approaches, we compare the performance against another metaheuristic considering both instances adapted from the literature as well as new instances based on a real case. Experiments using these instances ensure the validity of our approach and provide a reliable indication of the benefit of using our approach in real-world conditions.
- We provide decision-makers with insights into the effectiveness of using freight exchange platforms to reduce empty kilometers during return trips. Furthermore, we analyze the impact of various SMDPDPMTWPD problem variants on the routing schedules generated by our optimization approaches.

The remainder of this paper is organized as follows. A literature review is presented in Section 2. Then, Section 3 mathematically formulates the SMDPDPMTWPD. Section 4 describes our ALNS metaheuristic algorithm. Section 5 presents the numerical experiments, parameter tuning, and results. We end with conclusions and future research directions in Section 6.

2. Literature review

Pickup and delivery problems (PDPs) are a special category of vehicle routing problems (VRPs) where vehicles must fulfill a series of pickup and delivery requests [6]. There are several variants of the PDP, such as the Vehicle Routing Problem with Simultaneous Pickup and Delivery (VRPSPD), the Pickup and Delivery Vehicle Routing Problem with Time Windows (PDVRPTW), and the Pickup and Delivery Problem with Loading Constraints. This section reviews the relevant literature on PDPs that include key features of the SMDPDPMTWPD, such as multiple depots, customer selection, time windows, and pair demand. Section 2.1 reviews the literature on the Selective Full Truckload Multiple Depot Vehicle Routing Problem with Time Windows. Then, Section 2.2 reviews the literature on the Multiple Depot Pickup and Delivery Problem with Time Windows. After that, Section 2.3 reviews the literature on the Selective Pickup and Delivery Problem with Time Windows and Paired Demand. Finally, Section 2.4 presents the identified research gap that we aim to bridge with this paper.

2.1. The selective full truckload multiple depot vehicle routing problem with time windows

The Selective Full Truckload Multiple Depot Vehicle Routing Problem with Time Windows (SFTMDVRPTW) deals with empty return scenarios, where orders are selected and trucks are routed across multiple depots to deliver or pick up full truckloads while respecting customer time windows. The goal is to select a subset of Full Truckload (FTL) orders or customers, assign customers to vehicles, and find the optimal route for all vehicles that maximizes profit from serving customers while respecting time-window constraints. Each order has an associated revenue, cost, pickup and delivery location, and pickup and delivery time windows. In the SFTMDVRPTW, both the earliest departure and latest arrival time for each vehicle to respectively begin and complete its route are known in advance. Furthermore, transportation costs for empty kilometers and penalty costs for waiting are also known in advance for the case when a truck arrives early at a pickup or delivery location [5,7–11].

In the literature, we have identified a small research stream that focuses on the SFTMDVRPTW, which consists of six papers from the same authors [5,7–11]. We summarize the SFTMDVRPTW literature in Table 1. Overall, the authors treat the same mathematical model and solve instances of different sizes with their optimization approaches. Only one paper implements an exact approach with CPLEX to solve instances of up to thirty customers and three trucks [9]. El Bouyahyiouy and Bellabdaoui [9] solve to optimality twenty-two out of twenty-three instances within three minutes of computational time, achieving an optimality gap of 5.58% for the remaining instance. Different approximate approaches are implemented in the other papers: Ant Colony Optimization algorithm (ACO) [8], Genetic Algorithm (GA) [5,7,11], and a transformation in combination with Reactive Tabu Search (RTS) [10].

2.2. The multiple depot pickup and delivery problem with time windows

The Multiple Depot Pickup and Delivery Problem with Time Windows (MDPDPTW) deals with routing vehicles to serve customer orders, each consisting of a pickup and delivery request. Each pickup request is directly assigned to exactly one delivery request. Thus, the demand is paired, i.e., one-to-one [12]. Extensions of the MDPDPTW consider other types of assignations, where it is possible to have multiple pickup or delivery requests for a single commodity, i.e., one-to-many or many-to-many. The goal is to minimize the routing costs of vehicles starting from multiple depots such that all customers are served, while considering time windows, capacity, and precedence relations, i.e., a pickup location must be first visited before proceeding to its corresponding delivery location. According to Verdonck [13], the MDPDPTW has not

Table 1
Overview of the analyzed literature on the SFTMDVRPTW, with MD = Multiple Depots, F = Fleet (where HE = Heterogeneous and HO = Homogeneous), TW = Time Windows, C = Capacity, S = Selection, E = Exact, A = Approximate, N = Customers and K = Vehicles.

Reference	Problem features					Objective function	E/A	Solving method	Instance size (N, K)
	MD	F	TW	C	S				
El Bouyahyiouy and Bellabdaoui [7]	✓	HO	✓		✓	Max tot profit, consisting of (1) revenue, (2) moving costs, (3) empty costs and (4) waiting cost	A	GA	(13, 3)
El Bouyahyiouy and Bellabdaoui [8]	✓	HO	✓		✓	“”	A	ACO	(12, 2)
El Bouyahyiouy and Bellabdaoui [9]	✓	HO	✓		✓	“”	E	CPLEX	Up to (30, 3)
El Bouyahyiouy and Bellabdaoui [10]	✓	HO	✓		✓	“”	A	RTS	Up to (75, 7)
El Bouyahyiouy and Bellabdaoui [5]	✓	HO	✓		✓	“”	A	GA	Up to (75, 7)
El Bouyahyiouy and Bellabdaoui [11]	✓	HO	✓		✓	“”	A	GA + CPLEX	Up to (75, 7)
This work	✓	HE	✓	✓	✓	Max profit, consisting of (1) revenue, (2) moving costs, (3) driver costs and (4) serving costs	E & A	Solver & ALNS	Up to (250, 7)

been researched in depth. The relevant literature on the MDPDPTW is summarized in Table 2.

Dumas et al. [14] analyze the MDPDPTW and proposed an exact algorithm based on column generation (CG) and constrained shortest path to solve the pickup and delivery problem. The proposed algorithm can handle multiple depots and a heterogeneous vehicle fleet, showing a better performance if each customer has a large demand and the capacity constraints are restrictive. Then, Jung and Haghani [15] address the same problem but consider only one depot and an adapted objective function. Instead of only minimizing the routing costs, they also include the fixed costs of using the vehicles and customer inconvenience (penalty) costs due to the violation of the time windows. They assess their optimization approaches, showing that a Genetic Algorithm found the same optimal solutions as the MILP and outperforms the MILP for large instances. Additionally, a GA is also used in [16] to solve the MDPDPTW, where total route length is minimized.

Ropke and Pisinger [17] study an MDPDPTW variant with an objective function encompassing multiple cost aspects. In their MILP formulation, the objective function consists of a weighted sum of (1) the distance traveled, (2) the time spent by each vehicle, and (3) the number of requests not scheduled. This last element is particularly interesting, as [17] consider order selection in their model. Namely, some orders may be assigned to a virtual *request bank*, where an operator has to handle them. For instance, do nothing or hire extra vehicles. The model, however, does not distinguish between different orders and only includes penalty costs for the total number of orders not served. Thus, there is no profit or cost associated with each individual order. To solve the MDPDPTW variant, they developed an Adaptive Large Neighborhood Search (ALNS) heuristic, which was the first appearance of this method. In [13,18], the authors implement the same metaheuristic to study the collaboration between transportation companies. The authors of both papers formulate these problems as an MDPDPTW, where the routing costs are minimized and shared using game theoretic principles. The collaboration between transportation companies and the transformation of the problem to an MDPDPTW is also studied in [19], where the authors implement a different solving method, that is, a modified version of the heuristic-solver ROUTER.

Baldacci et al. [20] propose an exact algorithm to solve a single depot variant of the MDPDPTW. The algorithm is based on a set-partitioning-like integer formulation. The authors describe and implement bounding procedures that can find near-optimal dual solutions of the LP-relaxation. Heilig et al. [21] solve the multi-objective MDPDPTW implementing a Simulated Annealing (SA) algorithm. Although the main focus is on inter-terminal transport and not on the MDPDPTW, the authors argue that this problem can be modeled as an MDPDPTW, where the objective function consists of routing costs, delay penalties, and service times. Adi et al. [22] also focuses on inter-terminal transport but solves the problem using a different optimization approach: Reinforcement Learning (RL). They compare the performance of this approach with two other metaheuristic approaches from literature, i.e., Simulated Annealing and Tabu Search. They conclude that their RL algorithm outperforms the other approaches.

Finally, in [23,24], the authors consider the MDPDPTW to minimize the total route length and solve it using a Particle Swarm Optimization (PSO) algorithm. After testing the approaches on a large data set, both authors concluded that PSO outperforms other methods used in literature to solve these problem variants.

2.3. The selective pickup and delivery problem with time windows and paired demand

In the Selective Pickup and Delivery Problem with Time Windows and Paired Demands (SPDPTWPD) each customer order is associated with one pickup and delivery location. Thus, orders are paired one-to-one, meaning that each pickup request is exactly linked to one delivery request, and a given good can only be delivered if its pickup has occurred. The goal is to select a subset of customers and design routes for the vehicles to efficiently serve the selected customers while adhering to time windows, capacity, and precedence constraints. Table 3 summarizes the related literature on the SPDPTWPD.

The selective pickup and delivery problems are relatively recent, starting with Ting and Liao [25]. Although no time windows and paired orders are considered, the problem is still relevant to this review, as the model is similar to variants incorporating time windows and paired orders. That is, only few adaptions are necessary to incorporate the above-mentioned features. The problem is NP-hard, according to the authors, and, as a result, solved using a metaheuristic called Memetic Algorithm (MA). Ting et al. [26] propose a multi-vehicle version of the earlier model, with an additional restriction on the maximum travel distance. The model is solved using three different metaheuristics approaches, that is, Tabu Search (TS), Genetic Algorithm, and Scatter Search (SS). The results show that the Tabu Search outperforms the other two metaheuristics in solution quality and convergence speed.

Five papers from the same authors have appeared on the SPDPTWPD [27–31]. However, the objectives, solving methods, and mathematical formulations are different across the articles. Most of the models in these papers are solved utilizing the optimization solver Gurobi [27,28,31]. Furthermore, only two scientific papers are solved by implementing metaheuristics approaches, that is, Hybrid GA [29] and an extension of Tabu-embedded Simulated Annealing [30]. Additionally, single- and multi-objective functions are studied with profit maximization and distance minimization.

Besides that, the authors have also studied multi-period [32] and robust [33,34] versions of the problem, e.g., adding scenarios with uncertain traveling times, which are solved with Hybrid GA, Greedy Randomized Adaptive Search Procedure (GRASP), and a combination of GRASP with ALNS.

Table 2
Overview of the analyzed literature on the MDPDPTW, with MD = Multiple Depots, F = Fleet (where HE = Heterogeneous and HO = Homogeneous), TW = Time Windows, C = Capacity, S = Selection, E = Exact, A = Approximate, N = Customers and K = Vehicles.

Reference	Problem features					Objective function	E/A	Solving	Instance size
	MD	F	TW	C	S			method	(N, K)
Dumas et al. [14]	✓	HE	✓	✓		Min route costs	E	CG	Up to (55, 22)
Jung and Haghani [15]		HE	✓	✓		Min sum of route, usage and inconvenience costs	A	GA	Up to (30, 10)
Ropke and Pisinger [17]	✓	HE	✓	✓	✓	Min weighted sum of (1) distance traveled, (2) time spent by each vehicle and (3) requests not scheduled	A	ALNS	Up to (1000, 2759)
Krajewska et al. [18]	✓	HO	✓	✓		Min route costs	A	ALNS	Up to (250, 39)
Baldacci et al. [20]		HO	✓	✓		Min route costs w/wo fixed costs	E	Set partitioning + solver or algorithm	Up to (1000,100)
Dahl and Derigs [19]	✓	HE	✓	✓		Unknown	A	Modified version of ROUTER	Unknown
Alaïa et al. [16]	✓	HO	✓	✓		Min tot route length	A	GA	Up to (52, 24)
Alaïa et al. [23]	✓	HO	✓	✓		Min tot route length	A	PSO	Up to (53, 25)
Heilig et al. [21]	✓	HO	✓			Min sum of route, penalty and service times costs	A	SA	Up to (70, 17)
Verdonck [13]	✓	HO	✓	✓		Min route costs	A	ALNS	Up to (120, 5)
Harbaoui Dridi et al. [24]	✓	HO	✓	✓		Min total route length	A	PSO	Up to (53, 25)
Adi et al. [22]	✓	HO	✓	✓		Min usage, penalties, empty-truck and waiting costs	A	RL	Up to (285, 3)
This work	✓	HE	✓	✓	✓	Max profit, consisting of (1) revenue, (2) moving costs, (3) driver costs and (4) serving costs	E & A	Solver & ALNS	Up to (250, 7)

The authors of the previous papers have also contributed to other works [37,38]. In [37], the authors implemented a hybrid Particle Swarm Optimization (PSO), where the PSO is combined with local search procedures to solve a multiple objective model. The authors analyze a Pareto front for each instance, where profit maximization and distance minimization are optimized as objective functions. In [38], the authors focus on an extension called the Selective Pickup and Delivery Problem with Transfers (SPDPT). In the SPDPT, Transfers mean that goods can be transferred from one vehicle to another on special consolidation or transfer points. The model includes a max–min objective function, i.e., profit maximization and distance minimization, and is solved using a hybrid PSO.

Li et al. [35] study an extension of the SPDPTWPD, namely the Pickup and Delivery Problem with Time Windows, Profits, and Reserved Requests (PDPTWPR). This problem models carrier collaboration actions, where one part of the orders is reserved for each carrier, i.e., mandatory to be served. Then, a second part is selective, meaning that they may be served by the corresponding carrier or other carriers or completely rejected. To solve the PDPTWPR, the authors develop an ALNS metaheuristic with a meta-destroy mechanism and a dynamic adjustment of operator behavior (DAOB). Results show that ALNS finds the same optimal solutions as the MILP formulation for small instances of up to 50 requests, and outperforms the MILP formulation for large instances of up to 100 requests. Gansterer et al. [36] study another extension called the Multi-Vehicle Profitable Pickup and Delivery Problem (MVPPDP). The authors formulate the MVPPDP as a MILP and develop a metaheuristic framework based on General Variable Neighborhood Search (GVNS). They compare the GVNS performance to a Guided Local Search (GLS) based metaheuristic. Results show that the GVNS outperforms the GLS-based metaheuristic in terms of solution quality but at the expense of higher computational times.

Sun et al. [39] present a problem variant where travel times are dependent on the time of the day. This problem is called the Time-Dependent Profitable Pickup and Delivery Problem with Time Windows (TDPPDPTW). The authors argue that the work of [35,36] are the only papers covering similar mathematical models. The authors implement an ALNS metaheuristic to optimize profit maximization. The most recent scientific paper studying the TDPPDPTW is presented in [40], where both a PDPTWPR and a multi-objective SPDPTWPD are solved. The authors argue that the related literature mainly addresses single objectives and limited research is focused on solving bi-objective problem variants. The authors develop a two-phase heuristic framework (Pareto Local Search), based on the decomposition of the search space.

In addition to the previously discussed literature, we highlight the Team Orienteering Pickup and Delivery Problem with Time Windows (TOPDPTW) introduced in [41]. The TOPDPTW deploys a fleet of trucks to maximize the profit from selecting pickup and delivery orders, while adhering to hard time windows and capacity constraints. To solve this problem, the authors proposed a branch-and-price (BP) algorithm enhanced with a pruning technique, which significantly accelerates the solution of the associated subproblem, reducing computational times by 67%. Most research on the TOPDPTW either introduces new problem variants with similar characteristics or develops alternative BP algorithms with comparable features [42–44]. Furthermore, to the best of our knowledge, no metaheuristic algorithm has been proposed to solve larger instances for the TOPDPTW than those addressed in the original paper.

2.4. Research gap

Several similarities and differences emerge when comparing the key features of the analyzed transportation problems. Table 4 provides a comparative overview of the four problems most closely related and their relation to our work. To the best of our knowledge, no existing research examines an optimization problem that effectively addresses the logistics challenges that transportation companies face when using

Table 3
Overview of the analyzed literature on the SPDPTWPD, with MD = Multiple Depots, F = Fleet (where HE = Heterogeneous and HO = Homogeneous), TW = Time Windows, C = Capacity, S = Selection, E = Exact, A = Approximate, N = Customers and K = Vehicles.

Reference	Problem features					Objective function	E/A	Solving method	Instance size (N, K)
	MD	F	TW	C	S				
Ting and Liao [25]		HO		✓	✓	Min route costs	A	MA based on GA	Up to (500, 1)
Li et al. [35]		HE	✓	✓	✓	Max profit from selected orders	A	ALNS	Up to (100, ?)
Al Chami et al. [27]	✓	HE	✓	✓	✓	Max profit from or min distance of selected orders	E	Gurobi solver	Up to (100, 11)
Al Chami et al. [28]	✓	HE	✓	✓	✓	Max profit from and/or min distance of selected orders (both single and multi-obj)	E	Gurobi solver	Up to (100, 13)
Al Chami et al. [29]	✓	HE	✓	✓	✓	Max profit from and min distance of selected orders (multi-obj)	A	Hybrid GA	Up to (100, 11)
Ting et al. [26]		HO		✓	✓	Min route costs	A	TS, GA and SS	Up to (480, 22)
Gansterer et al. [36]		HO		✓	✓	Max profit from selected orders	S	GVNS	Up to (1000, 8)
Al Chami et al. [33]		HO	✓	✓	✓	Max profit from selected orders	A	GRASP	Up to (100, 11)
Al Chami et al. [32]		HE	✓	✓	✓	Min distance of selected orders	A	Hybrid GA	Up to 50 in 10 periods
Al Chami et al. [30]	✓	HE	✓	✓	✓	Max profit from and min the distance of selected orders (multi-obj)	A	Tabu SA	Up to (100, 13)
Al Chami et al. [31]	✓	HO	✓	✓	✓	Max profit from selected orders	E	Gurobi solver	Up to (100, 11)
Peng et al. [37]		HE	✓	✓	✓	Max profit from and min distance of selected orders (multi-obj)	A	Hybrid PSO	Up to (100, 13)
Peng et al. [38]		HE	✓	✓	✓	Max profit from and min distance of selected orders (multi-obj)	A	Hybrid PSO	Up to (100, ?)
Sun et al. [39]		HO	✓	✓	✓	Max profit from selected orders minus travel duration costs minus setup costs	A	ALNS	Up to (75, 4)
Al Chami et al. [34]		HO	✓	✓	✓	Max profit from selected orders	A	GRASP + ALNS	Up to (100, 11)
Ben-Said et al. [40]		HO	✓	✓	✓	Max profit from selected orders and min travel costs (multi-obj)	A	Two phase Pareto LS	Up to (100, 13)
This work	✓	HE	✓	✓	✓	Max profit, consisting of (1) revenue, (2) moving costs, (3) driver costs and (4) serving costs	E & A	Solver & ALNS	Up to (250, 7)

freight exchange platforms for return trips of empty trucks. The comparison in Table 4 highlights that while the reviewed problems share similarities with our research, none fully capture the problem addressed in this paper. Specifically, the SFTMDVRPTW considers only full truckloads, the MDPDPTW lacks order selection, and the SPDPTWPD does not account for multiple depots or different starting and ending depots. Likewise, the TOPDPTW excludes routing costs, preventing a realistic evaluation of the benefits of freight exchange platforms. Furthermore, none of these models incorporate multiple time windows for (un)loading or the possibility of multiple ending depots per truck. This research gap presents significant missed opportunities for freight exchange platform users, leading to limited competitiveness for small transport companies and inefficient freight matching in the market. Hence, we bridge this research gap by (*i*) defining this transportation problem as the SMDPDPMTWPD and proposing a mathematical formulation (see Section 3), and (*ii*) developing an ALNS metaheuristic framework (see Section 3) to efficiently solve large-scale instances within reasonable computational times.

3. The selective multiple depot pickup and delivery problem with multiple time windows and paired demand

This section formally introduces the SMDPDPMTWPD. A detailed description of the problem is provided in Section 3.1. Then, Section 3.2 enumerates the assumptions that a solution must satisfy. Finally, Section 3.3 presents a mathematical formulation for the SMDPDPMTWPD.

3.1. Problem description

The SMDPDPMTWPD is defined over a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, with node set $\mathcal{N}$ and arc set $\mathcal{A}$. The node set is defined as $\mathcal{N} = \mathcal{T} \cup \mathcal{P} \cup \mathcal{D} \cup \mathcal{T}'$ and the arc set as $\mathcal{A} = \{(i,j)|i \in \mathcal{T}, j \in \mathcal{P}\} \cup \{(i,j)|i,j \in \mathcal{P} \cup \mathcal{D} : i \neq j\} \cup \{(i,j)|i \in \mathcal{D}, j \in \mathcal{T}'\}$, where $\mathcal{T} = \{\tau_1, \dots, \tau_K\}$ is

Table 4
Comparison of the features of the reviewed model and the model of this research, with RC = Routing Costs, P = Profit, CO_2 = CO_2 emissions, C = Capacity, S = Selection, PD = Paired Demand, MLU = Mixed Loading/Unloading and Dif = Different.

Acronym	Objective			Depot			Time window		C	S	PD	MLU
	RC	P	CO_2	Multi	Dif start & end	Multi end	Single	Multi				
SFTMDVRPTW	✓	✓		✓	✓		✓			✓	✓	
MDPDPTW	✓			✓	✓		✓		✓		✓	✓
SPDPTWPD		✓					✓		✓	✓	✓	✓
TOPDPTW		✓			✓		✓		✓	✓	✓	✓
This work	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓

the set of starting depots, $\mathcal{T}' = \{\tau'_1, \dots, \tau'_K\}$ the set of ending depots, $\mathcal{P} = \{1, \dots, n\}$ is the set of pickup locations, $\mathcal{D} = \{n+1, \dots, 2n\}$ is the set of delivery locations, and $\mathcal{K} = \{1, \dots, K\}$ is the set of trucks. For simplicity, we define $\mathcal{Z} = \mathcal{T} \cup \mathcal{T}'$ as the depot set and $\mathcal{I} = \mathcal{P} \cup \mathcal{D}$ as the set of orders. Here, $\tau' = \{\tau'_{k1}, \dots, \tau'_{km_k}\}$ is the set of available ending depots for truck $k \in \mathcal{K}$, where m_k is the total number of ending depots for truck k. Furthermore, set $\mathcal{W}_i = \{1, \dots, M_i\}$ groups the available time windows at location $i \in \mathcal{N}$, where M_i is the total number of available time windows at location i.

The SMDPDPMTWPD consists of selecting orders from a large set of orders, e.g., freight exchange platforms, and determining optimal routes for a set of trucks $\mathcal{K}$ to maximize profit. Each truck $k \in \mathcal{K}$ has a maximum capacity Q^k and should execute the pickup and delivery operations at location $i \in \mathcal{I}$, within one of the $t \in \mathcal{W}_i$ the time-window intervals $[e_i^t, l_i^t]$ and with a duration of s_i. Early arrivals at the pickup and delivery locations are allowed, but vehicles should wait until the time window becomes available. Late arrival times are not allowed, except for the soft time-window variant. Thus, in the SMDPDPMTWPD, trucks should transport q_i goods from pickup locations $i \in \mathcal{P}$ to delivery locations $j \in \mathcal{D}$, where revenue p_i is earned, service costs c_i are incurred, and drivers' wages of f^t should be paid per unit of time. Additionally, for each arc $(i, j) \in A$ there is an associated travel cost c_{ij}, distance d_{ij}, and travel time t_{ij}.

The objective function optimizes the selection of orders and truck routing schedules to maximize profit, while respecting time-window constraints, maximum capacity, maximum weight, and precedence constraints, i.e., deliveries can only be executed after picking up the orders. The profit is calculated as the summation of all revenues minus the involved costs, consisting of: (*i*) transportation costs, (*ii*) service costs, and (*iii*) drivers' wages. Hence, in the SMDPDPMTWPD, the following decisions need to be made: (*i*) if vehicle $k \in \mathcal{K}$ travels from location $i \in \mathcal{N} \setminus \mathcal{T}'$ to $j \in \mathcal{N} \setminus \mathcal{T}$, (*ii*) if service at location $i \in \mathcal{I}$ is executed during time window $t \in \mathcal{W}_i$, (*iii*) waiting time of truck $k \in \mathcal{K}$ due to early arrival time at location $i \in \mathcal{N} \setminus \mathcal{T}$, (*iv*) load and weight on truck $k \in \mathcal{K}$ when leaving location $i \in \mathcal{N}$, and (*v*) arrival and departure time of truck $k \in \mathcal{K}$ at location $i \in \mathcal{N}$. Fig. 1 illustrates an example feasible routing schedule for the SMDPDPMTWPD.

The example illustrates a solution with $K = 3$ vehicles and $n = 5$ orders in the selection pool, e.g., orders available on the freight exchange platform, where depots are depicted by squares and the orders are represented by circles or nodes. The green and orange squares represent the start ($\mathcal{T}$) and ending ($\mathcal{T}'$) depots, respectively. Similarly, the blue and gray nodes show the selected and unselected orders. Additionally, diamonds represent the corresponding trucks, indicating the start and ending depots of each vehicle. Within the circles, the numbers represent the orders, whereas the plus (+) and minus (−) symbols show if they correspond to a pickup ($\mathcal{P}$) or delivery ($\mathcal{D}$) location. For clarity, the illustration does not display order weights, time windows, or vehicle capacity. However, it is important to note that a feasible solution must satisfy these constraints. Furthermore, each pickup and delivery location must have at least one time window.

In the solution, each vehicle departs from a different start location and may have multiple ending depots. For example, truck 1 can arrive at depots 9 or 10, while trucks 2 and 3 each have a single designated ending depot. Based on revenue and associated costs (travel, service, and driver wages), orders 4, 5, and 7 are selected for the return trips of trucks 1 and 2. Conversely, truck 3 travels directly from depot 3 to 10, as the remaining orders either fail to generate additional profit or cannot be accommodated due to time-window or capacity constraints. This shows the flexibility of the SMDPDPMTWPD in modeling real-world scenarios encountered by companies using freight exchange platforms. Hence, if an order is unprofitable or a truck cannot meet its constraints, the system ensures that the vehicle proceeds directly to its assigned depot. Additionally, trucks can pick up and deliver orders simultaneously, as seen in the route of truck 1. Instead of traveling directly from depot 1 to depot 9 or 10, truck 1 detours to pick up orders 4 and 7, maximizing the final profit.

3.2. Problem assumptions

The SMDPDPMTWPD is a transportation problem that models the challenge of using freight exchange platforms to reduce costs or maximize profits of empty trucks in their return trips. Consequently, to properly model these transportation dynamics, the fundamental assumptions of the SMDPDPMTWPD are outlined as follows:

- An order can only be selected if the time window constraints are satisfied for both pickups and deliveries.
- Each order has one pickup and one delivery location, which can be served by at most one vehicle.
- Direct paired orders are considered. Each pickup location is directly paired with a specific delivery location. Thus, paired orders must be handled by the same vehicle.
- Sequential pickup and delivery operations are allowed. Hence, in the same trip, trucks should first pickup the goods before delivering them.
- Each vehicle departs from one starting depot and may have one or several possible ending depots to arrive.
- Each vehicle can only start from its starting depot after the starting time has passed. Similarly, a vehicle must arrive at one of the ending depots before its corresponding closing time.
- Vehicles can arrive earlier than the opening time of the time windows at the pickup and delivery locations. However, vehicles have to wait until the time windows become available to perform the service.
- A vehicle k cannot load more than its maximum capacity Q^k. The capacity Q^k is measured in Loading Meters (LDM; see explanation below).

We express the capacity of trucks in loading meters (LDM). LDMs are a common measurement unit in road transport for a freight that is not stackable [45]. One LDM is equivalent to 1.00 m length from the back of a trailer. A standard trailer has a width of 2.4 m and, hence, one LDM is equivalent to $1.00 \times 2.40 = 2.40$ m^2 [46]. The LDM of freights is calculated as shown in Eq. (1).

$$\text{LDM} = \frac{(\text{length of freight} \times \text{width of freight})}{\text{width of trailer}} \tag{1}$$

Measuring the capacity of trucks in LDM generalizes the capacity across different freights. Most freights are transported on Europallets, with dimensions 1.20 by 0.80 m. A Europallet is, therefore, equivalent to $(1.20 \times 0.80)/2.40 = 0.40$ LDMs. A standard truck with a capacity of 13.60 LDM can, thus, carry a total of $13.60/0.40 = 34$ Europallets.

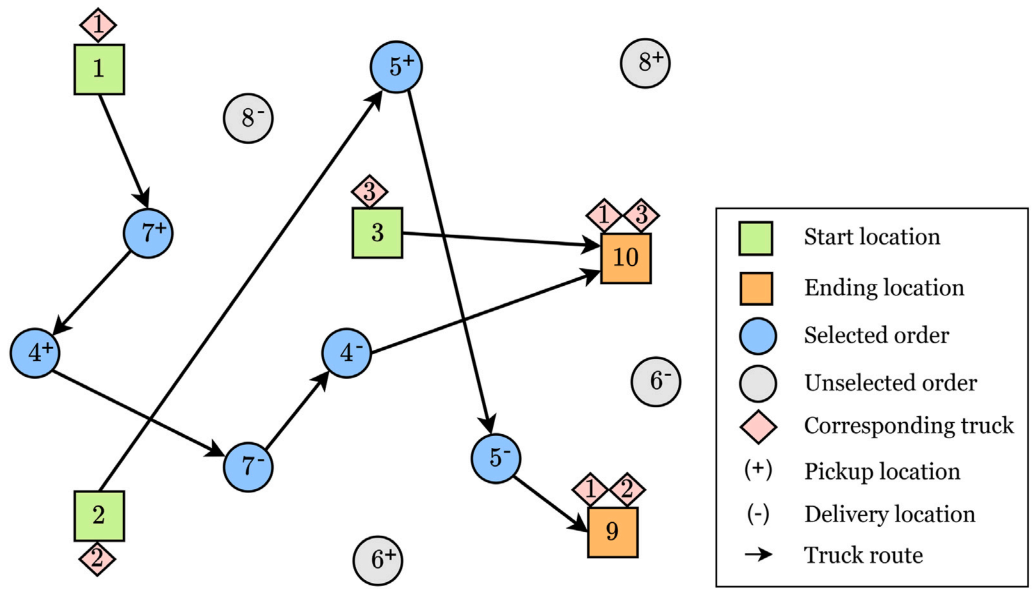

Fig. 1. Illustration of a feasible solutions for the SMDPDPMTWPD.

3.3. Mathematical model

The profit maximization SMDPDPMTWPD is formulated as a MILP in Section 3.3.1. Then, Section 3.3.2 extends the SMDPDPMTWPD to study the profit maximization including costs for CO_2 emissions. After that, Section 3.3.3 introduces the soft time-window variant. Finally, Section 3.3.4 presents an extension that considers soft time windows and costs for CO_2 emissions.

3.3.1. The profit maximization SMDPDPMTWPD

Table 5 presents the sets and parameters for the MILP formulation of the profit maximization SMDPDPMTWPD, and the extensions presented in Sections 3.3.2, 3.3.3, and 3.3.4.

The following decision variables are involved:

$$x_{ij}^k = \begin{cases} 1, & \text{if vehicle } k \in \mathcal{K} \text{ travels from location} \\ & i \in \mathcal{N}\setminus\mathcal{T}' \text{ to location } j \in \mathcal{N}\setminus\mathcal{T} \\ 0, & \text{otherwise} \end{cases}$$

$$b_i^t = \begin{cases} 1, & \text{if service takes place during time window} \\ & t \in \mathcal{W}_i \text{ at location } i \in \mathcal{N} \\ 0, & \text{otherwise} \end{cases}$$

w_i = waiting time (earliness) before picking up or delivering at location $i \in \mathcal{N}$

q_i^k = load of vehicle $k \in \mathcal{K}$ when leaving from location $i \in \mathcal{N}$

h_i^k = weight of vehicle $k \in \mathcal{K}$ when leaving from location $i \in \mathcal{N}$

a_i^k = arrival time of vehicle $k \in \mathcal{K}$ at location $i \in \mathcal{N}$

d_i^k = departure time of vehicle $k \in \mathcal{K}$ at location $i \in \mathcal{N}$

y_i^k = arrival time of vehicle $k \in \mathcal{K}$ at location $i \in \tau_k'$

The MILP formulation of the profit maximization SMDPDPMTWPD is as follows:

$$Max \quad Z = \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{N}\setminus\mathcal{T}'} \sum_{j\in\mathcal{N}\setminus\mathcal{T}\cup\{i\}} \left(p_j - c_j - c_{ij}\right) x_{ij}^k - \sum_{k\in\mathcal{K}} \sum_{i\in\tau'} f^t y_i^k \tag{2}$$

$$\text{s.t.} \quad \sum_{k\in\mathcal{K}} \sum_{\substack{j\in\mathcal{N}\setminus\mathcal{T}, \\ j\neq i}} x_{ij}^k \leq 1 \qquad i \in \mathcal{P} \tag{3}$$

$$\sum_{\substack{j\in\mathcal{N}\setminus\mathcal{T}, \\ j\neq i}} x_{ij}^k = \sum_{\substack{j\in\mathcal{N}\setminus\mathcal{T}, \\ j\neq i}} x_{(n+i)j}^k \qquad i \in \mathcal{P}, k \in \mathcal{K} \tag{4}$$

$$\sum_{\substack{j\in\mathcal{N}\setminus\mathcal{T}', \\ j\neq i}} x_{ji}^k = \sum_{\substack{j\in\mathcal{N}\setminus\mathcal{T}, \\ j\neq i}} x_{ij}^k \qquad i \in \mathcal{I}, k \in \mathcal{K} \tag{5}$$

$$x_{ii}^k = 0 \qquad i \in \mathcal{N}, k \in \mathcal{K} \tag{6}$$

$$\sum_{j\in\mathcal{P}\cup\tau_k'} x_{\tau_k j}^k = 1 \qquad k \in \mathcal{K} \tag{7}$$

$$\sum_{i\in\mathcal{T}} \sum_{j\in\mathcal{N}\setminus\mathcal{T}} x_{ij}^k = 1 \qquad k \in \mathcal{K} \tag{8}$$

$$\sum_{i\in\mathcal{D}\cup\tau_k} \sum_{z\in\tau_k'} x_{iz}^k = 1 \qquad k \in \mathcal{K} \tag{9}$$

$$\sum_{i\in\mathcal{N}\setminus\mathcal{T}'} \sum_{j\in\mathcal{T}'} x_{ij}^k = 1 \qquad k \in \mathcal{K} \tag{10}$$

$$d_i^k \geq a_i^k \qquad i \in \mathcal{N}, k \in \mathcal{K} \tag{11}$$

$$a_{(n+i)}^k + w_{(n+i)} \geq a_i^k + w_i + s_i + t_{i,(n+i)} - M\left(1 - \sum_{j\in\mathcal{N}\setminus\mathcal{T}} x_{ij}^k\right) \qquad i \in \mathcal{P}, k \in \mathcal{K} \tag{12}$$

$$d_{\tau_k}^k = \sum_{t\in\mathcal{W}_{\tau_k}} e_{\tau_k}^t b_{\tau_k}^t \qquad k \in \mathcal{K} \tag{13}$$

$$y_j^k \geq a_j^k - a_{\tau_k}^k - M\left(2 - \sum_{z\in\mathcal{P}\cup\tau_k'} x_{\tau_k z}^k - \sum_{z\in\mathcal{D}\cup\tau_k} x_{zj}^k\right) \qquad j \in \tau_k', k \in \mathcal{K} \tag{14}$$

$$a_j^k \geq d_i^k + t_{ij} - M\left(1 - x_{ij}^k\right) \qquad i \in \mathcal{N}\setminus\mathcal{T}', j \in \mathcal{N}\setminus\mathcal{T}, k \in \mathcal{K} : i \neq j \tag{15}$$

$$a_j^k \leq a_i^k + w_i + s_i + t_{ij} + M\left(1 - x_{ij}^k\right) \qquad i \in \mathcal{N}\setminus\mathcal{T}', j \in \mathcal{N}\setminus\mathcal{T}, k \in \mathcal{K} : i \neq j \tag{16}$$

$$d_j^k \geq d_i^k + t_{ij} + w_j + s_j - M\left(1 - x_{ij}^k\right) \qquad i \in \mathcal{N}\setminus\mathcal{T}', j \in \mathcal{N}\setminus\mathcal{T}, k \in \mathcal{K} : i \neq j \tag{17}$$

$$w_j \geq \sum_{t\in\mathcal{W}_j} e_j^t b_j^t - \left(d_i^k + t_{ij} + M\left(1 - x_{ij}^k\right)\right) \qquad i \in \mathcal{N}\setminus\mathcal{T}', j \in \mathcal{N}\setminus\mathcal{T}, k \in \mathcal{K} : i \neq j \tag{18}$$

$$d_i^k + t_{ij} + w_j + s_j - M(1 - x_{ij}^k) \leq \sum_{t\in\mathcal{W}_j} l_j^t b_j^t \qquad i \in \mathcal{N}\setminus\mathcal{T}', j \in \mathcal{N}\setminus\mathcal{T}, k \in \mathcal{K} : i \neq j \tag{19}$$

Table 5
Sets and parameters for the MILP formulation of the profit maximization SMDPDPMTWPD and its extensions.

Parameters		Sets	
n	Number of orders	$\mathcal{G}$	Direct graph, $\mathcal{G} = \{\mathcal{N}, \mathcal{A}\}$
K	Number of vehicles	$\mathcal{N}$	Node set, $\mathcal{N} = \{\mathcal{I}, \mathcal{Z}\}$
Q^k	Capacity of vehicle k (in LDMs)	$\mathcal{A}$	Arc set, $\mathcal{A} = \{(i,j) \mid i \in \mathcal{T}, j \in \mathcal{P}\} \cup \{(i,j) \mid i,j \in \mathcal{P} \cup \mathcal{D} : i \neq j\} \cup \{(i,j) \mid i \in \mathcal{D}, j \in \mathcal{T}'\}$
H^k	Maximum weight allowed in vehicle k (in kg's)		
h^k_{start}	Start weight of vehicle k (in kg's)		
$[e^t_i, l^t_i]$	Time window t of location i, where e^t_i and l^t_i are the time (in minutes) on the custom clock	$\mathcal{I}$	Order set, $\mathcal{I} = \mathcal{P} \cup \mathcal{D} = \{1, \dots, 2n\}$
M	Big number	$\mathcal{P}$	Pickup set, $\mathcal{P} = \{1, \dots, n\}$
M_i	Numbers of time windows during which (un)loading may take place at location i	$\mathcal{D}$	Delivery set, $\mathcal{D} = \{n+1, \dots, 2n\}$
s_i	Service time of an order to be (un)loaded at location i, where $s_i = 0$ for a depot and $s_i \geq 0$ for any other location	$\mathcal{Z}$	Depot set, $\mathcal{Z} = \mathcal{T} \cup \mathcal{T}'$
q_i	Demand of an order to be (un)loaded at location i, where $q_i \geq 0$ for a pickup location, $q_i \leq 0$ for a delivery location and $q_i = 0$ for a depot	$\mathcal{T}$	Start location set, $\mathcal{T} = \{\tau_1, \dots, \tau_K\}$
h_i	Weight of an order to be (un)loaded at location i, where $h_i \geq 0$ for a pickup location, $h_i \leq 0$ for a delivery location and $h_i = 0$ for a depot	$\mathcal{T}'$	End location set, $\mathcal{T}' = \{\tau'_1, \dots, \tau'_K\}$, where $\tau'_k = \{\tau'_{k1}, \dots, \tau'_{km_k}\}$
c_i	Costs of an order to be (un)loaded associated with (service on) location i, where $c_i \geq 0$ for a pickup or delivery location and $c_i = 0$ for a depot	$\mathcal{W}_i$	Index set of time windows at location i, where $\mathcal{W}_i = \{1, \dots, M_i\}$
p_i	Revenue of an order to be (un)loaded location i, where $p_i \geq 0$ for a pickup location (representing the revenue associated with an order) and $p_i = 0$ for a delivery location or a depot	$\mathcal{K}$	Vehicle set, $\mathcal{K} = \{1, \dots, K\}$
d_{ij}	Distance between location i and j		
c_{ij}	Cost of traveling between location i and j		
t_{ij}	Travel time between location i and j		
f^t	Wage of a truck driver per minute		
m_k	Number of ending locations for vehicle k		

$$\sum_{t \in \mathcal{W}_i} b^t_i = 1 \qquad i \in \mathcal{Z} \qquad (20)$$

$$\sum_{t \in \mathcal{W}_i} b^t_i = \sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{N} \setminus \mathcal{T}'} x^k_{ji} \qquad i \in \mathcal{I} \qquad (21)$$

$$q^k_j \geq q^k_i + q_j - M\left(1 - x^k_{ij}\right) \qquad i \in \mathcal{N} \setminus \mathcal{T}', j \in \mathcal{N} \setminus \mathcal{T}, k \in \mathcal{K} : i \neq j \qquad (22)$$

$$q^k_i \leq Q^k \sum_{\substack{j \in \mathcal{N} \setminus \mathcal{T}, \\ j \neq i}} x^k_{ij} \qquad i \in \mathcal{P}, k \in \mathcal{K} \qquad (23)$$

$$q^k_i \leq \left(Q^k + q_i\right) \sum_{\substack{j \in \mathcal{N} \setminus \mathcal{T}, \\ j \neq i}} x^k_{ij} \qquad i \in \mathcal{D}, k \in \mathcal{K} \qquad (24)$$

$$h^k_j \geq h^k_i + h_j - M\left(1 - x^k_{ij}\right) \qquad i \in \mathcal{N} \setminus \mathcal{T}', j \in \mathcal{N} \setminus \mathcal{T}, k \in \mathcal{K} : i \neq j \qquad (25)$$

$$h^k_i \leq \left(H^k - h^k_{start}\right) \sum_{\substack{j \in \mathcal{N} \setminus \mathcal{T}, \\ j \neq i}} x^k_{ij} \qquad i \in \mathcal{P}, k \in \mathcal{K} \qquad (26)$$

$$h^k_i \leq \left(H^k - h^k_{start} + h_i\right) \sum_{\substack{j \in \mathcal{N} \setminus \mathcal{T}, \\ j \neq i}} x^k_{ij} \qquad i \in \mathcal{D}, k \in \mathcal{K} \qquad (27)$$

$$x^k_{ij} \in \{0, 1\} \qquad i, j \in \mathcal{N}, k \in \mathcal{K} \qquad (28)$$

$$b^t_i \in \{0, 1\} \qquad i \in \mathcal{N}, t \in \mathcal{W}_i \qquad (29)$$

$$a^k_i, d^k_i, y^k, w_i, q^k_i, h^k_i \geq 0 \qquad i \in \mathcal{N}, k \in \mathcal{K} \qquad (30)$$

The objective function (2) maximizes the profit earned from the selected orders. The first summation represents the revenue obtained from the selected orders, whereas the second summation represents the driver wages associated with the total transportation times. Constraints (3) state that every pickup location can be visited at most once. Constraints (4) ensure that the same truck performs the pickup and delivery of a given order. Constraints (5) state that if a truck visits a certain location, the same truck should leave from it. Constraints (6) prevent cycles in the truck routes. Constraints (7) force the trucks to depart from the starting depots. Constraints (8) establish that each truck is utilized only once. Constraints (9) ensure that each truck arrives at only one of its multiple ending depots. Constraints (10) in combination with Constraints (9) force the trucks not to visit the arriving depots of other trucks. Constraints (11) state that the departure time of trucks from a given node is greater or equal to the arrival time of the trucks. Constraints (12) ensure that, for a given truck, all the deliveries are executed after picking them up. Constraints (13) impose that the trucks' departure times are equal to the opening times of the time windows from where they depart. Constraints (14) define the total routing time of each truck. Constraints (15) establish that, when traveling from i to j, the truck arrival time at j is greater than or equal to the departure time from i plus the transportation times between i and j. Constraints (16) link the arrival times with the trucks' waiting times when traveling from i to j. Constraints (17) link the departure times with the trucks' waiting times when traveling from i to j. Constraints (18) define the trucks' waiting times. Constraints (19) establish that the delivery and pickups should be executed before the closing times of the time windows. Constraints (20) ensure that only one of the time windows is chosen for every starting and ending depot. Constraints (21) indicate that only one of the multiple time windows must be chosen at each customer location. Constraints (22)–(24) establish the flow balance of the loads on the truck routes. Constraints (25)–(27) ensure the flow balance of the weights on the truck routes. Constraints (28)–(30) present the variable definitions.

3.3.2. The profit maximization including CO_2 emissions SMDPDPMTWPD

This problem variant includes CO_2 emissions in the profit maximization objective function. We follow the approach from [47] to transform the CO_2 emissions of going from location i to j into a cost factor.

In this approach, an emission market price c_e (€$/kg\,CO_2$) is used for the transformation of emissions into costs. Additionally, we utilize a fuel consumption estimation method that calculates the CO_2 emitted between two locations based on the traveled distance d_{ij} (km) and a conversion factor α ($kg\,CO_2/km/LDM$) [48]. Therefore, we calculate the CO_2 emissions costs by multiplying c_e, α, and d_{ij} by the total truck weight when traversing arc $(i,j) \in \mathcal{A}$. Consequently, to include the CO_2 emissions cost factor into the objective function, we introduce a new variable z_{ij}^k, defined as the total weight of truck $k \in \mathcal{K}$, when traversing arc $(i,j) \in \mathcal{A}$. Due to the multiplication of α by the number of kilometers and the load of the vehicles, this model variant belongs to the Cumulative Vehicle Routing Problems class, see [49]. The MILP formulation of the profit maximization with CO_2 emissions SMDPDPMTWPD is defined as follows:

$$Max \quad Z = \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{N}\setminus\mathcal{T}'} \sum_{j\in\mathcal{N}\setminus\mathcal{T}\cup\{i\}} \left(\left(p_j - c_j - c_{ij}\right) x_{ij}^k - \left(c_e \cdot \alpha \cdot d_{ij}\right) z_{ij}^k \right) - \sum_{k\in\mathcal{K}} \sum_{i\in\tau'} f^t y_i^k \tag{31}$$

s.t. (3)–(30)

$$z_{ij}^k \le M\ x_{ij}^k \qquad i\in\mathcal{N}\setminus\mathcal{T}', j\in\mathcal{N}\setminus\mathcal{T}, k\in\mathcal{K} : i\neq j \tag{32}$$

$$z_{ij}^k \le q_i^k \qquad i\in\mathcal{N}\setminus\mathcal{T}', j\in\mathcal{N}\setminus\mathcal{T}, k\in\mathcal{K} : i\neq j \tag{33}$$

$$z_{ij}^k \ge q_i^k - M\left(1 - x_{ij}^k\right) \qquad i\in\mathcal{N}\setminus\mathcal{T}', j\in\mathcal{N}\setminus\mathcal{T}, k\in\mathcal{K} : i\neq j \tag{34}$$

$$z_{ij}^k \ge 0 \qquad i\in\mathcal{N}\setminus\mathcal{T}', j\in\mathcal{N}\setminus\mathcal{T}, k\in\mathcal{K} : i\neq j \tag{35}$$

The objective function (31) optimizes the profit maximization SMDPDPMTWPD with CO_2 emissions costs. Constraints (3)–(30) are directly extracted from the MILP formulation in Section 3.3.1. Additionally, Constraints (32)–(35) define vehicles' total weight on the network. Constraint (32) link the z_{ij}^k and x_{ij}^k variables. Thus, vehicles can only have a weight assigned on those arcs that are traversed by them. Constraints (33) establish that when traversing arc (i,j), the total weight on the vehicle should be less or equal to the weight q_i^k, after leaving location i. Constraints (34) ensures that when a given vehicle traverses arc (i,j), the total weight on the vehicle is at least equal to the weight q_i^k, after leaving location i. Finally, the definition of variable z_{ij}^k is presented by Constraints (35).

3.3.3. The soft time windows SMDPDPMTWPD

The soft time windows SMDPDPMTWPD allows vehicles to perform late deliveries, i.e., trucks can execute deliveries after the closing time of the time windows. In those cases where vehicles arrive early, trucks still have to wait until the opening time of the time windows. To model this problem extension, we adjust the objective function by incorporating a penalization for late deliveries c^l. Consequently, to compute this penalization, we introduce an additional variable m_i, defined as the delayed delivery time at a location $i \in \mathcal{N}$. Furthermore, we drop Constraints (19) to allow late deliveries. The MILP formulation of the soft time windows SMDPDPMTWPD is defined as follows:

$$Max \quad Z = \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{N}\setminus\mathcal{T}'} \sum_{j\in\mathcal{N}\setminus\mathcal{T}\cup\{i\}} \left(p_j - c_j - c_{ij}\right) x_{ij}^k - \sum_{k\in\mathcal{K}} \sum_{i\in\tau'} f^t y_i^k - \sum_{i\in\mathcal{N}} c^l m_i \tag{36}$$

s.t. (3)–(18), (20)–(30)

$$m_j \ge d_i^k + t_{ij} + s_i - M\left(1 - x_{ij}^k\right) - \sum_{t\in\mathcal{W}_j} l_j^t b_j^t \qquad i\in\mathcal{N}\setminus\mathcal{T}', j\in\mathcal{N}\setminus\mathcal{T}, k\in\mathcal{K} : i\neq j \tag{37}$$

$$m_j \ge 0 \qquad i\in\mathcal{N}, k\in\mathcal{K} \tag{38}$$

The objective function (36) maximizes the profit for the soft time windows SMDPDPMTWPD. Constraints (3)–(18) and (20)–(30) are directly extracted from the MILP formulation present in Section 3.3.1. Then, Constraints (37) compute the delayed delivery time at location i, as the difference between the closing time of the selected time window b_i^t and the ending time of the service, i.e., $d_i^k + t_{ij} + s_i$. Finally, Constraints (37) provide the definition of variables m_i.

3.3.4. The soft time windows including CO_2 emissions SMDPDPMTWPD

The last problem variant combines soft time windows, profit maximization, and CO_2 emissions costs. The MILP formulation is a combination of the MILPs introduced in Sections 3.3.1–3.3.3. The MILP formulation of the profit maximization SMDPDPMTWPD with soft time windows and CO_2 emissions costs is as follows:

$$Max \quad Z = \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{N}\setminus\mathcal{T}'} \sum_{j\in\mathcal{N}\setminus\mathcal{T}\cup\{i\}} \left(\left(p_j - c_j - c_{ij}\right) x_{ij}^k - \left(c_e \cdot \alpha \cdot d_{ij}\right) z_{ij}^k \right) - \sum_{k\in\mathcal{K}} \sum_{i\in\tau'} f^t y_i^k - \sum_{i\in\mathcal{N}} c^l m_i \tag{39}$$

s.t. (3)–(18), (20)–(30), (32)–(35), (37)–(38)

4. Solution approach

As a generalization of the VRP, the SMDPDPMTWPD is an NP-hard problem [50], meaning that the SMDPDPMTWPD is at least as complex as the VRP. Given the computational hardness of the SMDPDPMTWPD, we develop a metaheuristic approach to solve it. Section 4.1 presents the ALNS framework developed for solving the SMDPDPMTWPD. Section 4.2 describes the methods for constructing the initial solution. The destroy and repair operators are detailed in Sections 4.3 and 4.4, respectively. Finally, Section 4.5 outlines the feasibility checks applied to each new solution to ensure its feasibility.

4.1. Adaptive large neighborhood search

Adaptive Large Neighborhood Search (ALNS) is a relatively new metaheuristic introduced in [17], designed to explore the solution space using multiple destroy (Ω^d) and repair (Ω^r) operators. ALNS dynamically adjusts operator selection, assigning higher probabilities to more effective operators while limiting the use of less effective ones. This adaptability is one of ALNS's key advantages, improving search efficiency by focusing on the most promising strategies. In this research, we develop an ALNS metaheuristic to solve the SMDPDPMTWPD, as our literature review (see Section 2) indicates that ALNS has been highly effective in solving large-scale instances, outperforming other approximate methods for similar transportation problems (see Table 2).

Algorithm 1 describes the ALNS metaheuristic framework considered in this paper. The input parameters to ALNS are the starting T_{start} and ending T_{end} temperatures, the cooling ratio α, the maximum number of rejected-orders for creating or repairing a solution r, the number of iterations N_{iter} at a given temperature, the learning coefficient η, the destruction factor ρ, and the maximum computational time δ_T. The output is the best solution found during the exploration, i.e., the incumbent x^*. In the algorithm, the current solution is denoted by x, the created neighbor after applying the selected destroy and repair operators is given by x', and the objective function value for any solution is determined by $f(\cdot)$. Furthermore, to execute the exploration of the solution space, ALNS utilizes vectors that represent the weights of the destroy ω^d and repair ω^r operators, probabilities to select destroy p^d and repair p^r operators, utilization counters of destroy α^d and repair α^r operators, and success rates of the destroy β^d and repair operators β^r. All these vectors incorporate an additional subindex to refer to a specific operator, e.g., ω_a^d denotes the weight of the destroy operator $a \in \Omega^d$. The reader is referred to Section 5.1 for further information on parameter tuning of ALNS.

Algorithm 1: ALNS

Data: $(T_{start}, T_{end}, \alpha, r, N_{iter}, \eta, \rho, \delta_T)$

1 **Initialization Phase:**

2 $T \leftarrow T_{start}$;

3 $time_{init}, time_{end} \leftarrow time()$;

4 $\omega^d, \omega^r, p^d, p^r, \alpha^d, \alpha^r, \beta^d, \beta^r \leftarrow$ **Initialize(** $\cdot$ **)**;

5 $x, x^* \leftarrow$ **Constructive_Heuristic(**r**)**;

6 **Iterative Phase:**

7 **while** $time_{end} - time_{init} < \delta_T$ *and* $T > T_{end}$ **do**

8 **for** N_{iter} **do**

9 $a \leftarrow \Omega^d(p^d)$;

10 $b \leftarrow \Omega^r(p^r)$;

11 $x' \leftarrow$ **Construct_Neighbor(**$x, a, b, \lceil \rho S \rceil, r$**)**;

12 $\alpha^d_a \leftarrow \alpha^d_a + 1$ and $\alpha^r_b \leftarrow \alpha^r_b + 1$;

13 **if** $f(x') > f(x)$ **then**

14 $x \leftarrow x'$;

15 $\beta^d_a \leftarrow \beta^d_a + \sigma_2$ and $\beta^r_b \leftarrow \beta^r_b + \sigma_2$;

16 **if** $f(x') > f(x^*)$ **then**

17 $x^* \leftarrow x'$;

18 $\beta^d_a \leftarrow \beta^d_a + \sigma_1$ and $\beta^r_b \leftarrow \beta^r_b + \sigma_1$;

19 **else**

20 $\phi \leftarrow rand(0,1)$;

21 **if** $\phi < e^{\frac{(f(x')-f(x))}{T}}$ **then**

22 $x \leftarrow x'$;

23 $\beta^d_a \leftarrow \beta^d_a + \sigma_3$ and $\beta^r_b \leftarrow \beta^r_b + \sigma_3$;

24 $T \leftarrow \alpha \cdot T$;

25 $\omega^d, \omega^r \leftarrow$ **Update_Weights(**$\omega^d, \omega^r, \eta, \alpha^d, \alpha^r, \beta^d, \beta^r$**)** ;

26 $p^d, p^r \leftarrow$ **Update_Probabilities(**ω^d, ω^r**)** ;

27 $time_{end} \leftarrow time()$;

Result: x^*

The algorithm starts with the initiation phase (lines 1–5), by setting up the initial temperature (line 2), establishing the time handlers $time_{init}$ and $time_{end}$ (line 3), initializing the parameters of the destroy and repair operators (line 4), and creating the initial solution and storing it in x and x^* (line 5). Subsequently, the iterative phase is applied until the stopping criterion is met (lines 6–27). ALNS stops the exploration when either (*i*) the executed time is equal or higher than δ_T, or (*ii*) T is equal to or less than T_{end} (line 6). In the while-loop, a for-loop is applied to create new neighboring solutions by applying the destroy and repair operators (lines 8–23). In every iteration of the for-loop, destroy and repair operators a and b are selected (lines 9–10). The selection of the destroy and repair operators is based on the roulette wheel selection method, utilizing the probability vectors p^d and p^r, respectively. Then, a new neighboring solution is created and stored in x' (line 11). The extent to which a solution is destroyed depends on the destruction rate $\lceil \rho S \rceil$, where S is the total number of orders in the current solution and ρ is a removal fraction $\in [0,1]$. Next, a repair operator is applied until the list containing the rejected orders is at least equal to r. Subsequently, the utilization counters of the selected destroy and repair operators a and b are increased by one unit (line 12). The new solution x' is evaluated and accepted according to the SA acceptance criterion (lines 13–23). If the solution x' outperforms x, i.e., $f(x') > f(x)$, the new solution x' is stored in x and the success rates β^d_a and β^r_b are increased by σ_2 (lines 13–15). In addition to this, if the new solution outperforms the incumbent, i.e., $f(x') > f(x^*)$, the incumbent is updated, and the success rates are updated according to σ_1 (lines 16–18). Otherwise, if the newly created neighbor x' does not outperform x, the acceptance criterion based on the annealing temperature is applied (lines 19–23). Here, a random number from the interval $(0,1)$ is chosen and stored in ϕ (line 20). Then, if the inequality $\phi < e^{\frac{(f(x')-f(x))}{T}}$ holds, x' is accepted and stored in x, and the success rates β^d_a and β^r_b are updated according to σ_3 (lines 21–23). In this ALNS implementation, σ_1, σ_2, and σ_3 represent the success rates utilized to update β^d and β^r, where $\sigma_1 > \sigma_2 > \sigma_3$. Subsequently, once the algorithm leaves the for-loop, the current temperature is decreased according to the cooling ratio (line 24), the weights of the destroy and repair operators are updated based on Eq. (40) (line 25), the selection probabilities p^d and p^r are updated according to Eq. (41) (line 26), and the time handler $time_{end}$ is updated (line 27). For simplicity, indexes r and d are skipped in Eqs. (40) and (41). Finally, the algorithm returns the best solution found during the exploration, i.e., the incumbent x^*.

$$\omega_i = \omega_i \cdot (1-\eta) + \eta \cdot \frac{\beta_i}{\alpha_i} \qquad i \in \Omega \tag{40}$$

$$p_i = \frac{\omega_i}{\sum_{j\in\Omega} \omega_j} \qquad i \in \Omega \tag{41}$$

4.2. Initial solution

This section describes the procedure of the **Constructive_Heuristic**$(\cdot)$ function implemented in the ALNS metaheuristic framework (see Algorithm 1). This heuristic implements a random generation procedure to build initial routes.

The random generation procedure starts by creating an empty route for every vehicle consisting only of the starting and ending depot. If a given vehicle has multiple ending depots, one of them is randomly chosen. Then, for each vehicle, a random order consisting of the pickup and delivery location is assigned. Orders are randomly located along the routes, where the pickup and delivery locations of an order are placed consecutively behind each other. If the solution is feasible, the order is accepted in the route. Otherwise, the order is not accepted and is stored in a list of rejected orders. This process is repeated until the length of the rejected-order list of each vehicle is equal to r. We have established this stopping criterion to prevent large computational times.

4.3. Destroy operators

We selected five destroy operators in the ALNS metaheuristic. These destroy operators are based on [17,35,39], and [51], which are listed as follows:

- **Random removal**: This operator is the most basic destroy operator, which randomly selects some orders (the pickup and delivery locations) and deletes them from the current solution.
- **Shaw removal**: In this removal, similar requests are selected and removed from the current solution. The idea is that similar requests can easily be shuffled. Thus a new feasible solution, possibly better than the previous one, is created. Similar requests have a low similarity score sim_{ij}, calculated as shown in Eq. (42). Here, d_{ij} represents the distance between two locations, s^k_i is the service time at a given location i, q_i is the load of the vehicle when leaving location i, and $\lambda_d + \lambda_t + \lambda_q = 1$ with values $\lambda_d = \lambda_t = \lambda_q = 1/3$. The plus (+) and minus (−) symbols in the subscripts indicate a pickup and delivery location, respectively.

$$sim_{ij} = \lambda_d * \left(d_{i^+j^+} + d_{i^-j^-}\right) + \lambda_t * \left(|s^k_{i^+} - s^k_{j^+}| + |s^k_{i^-} - s^k_{j^-}|\right) + \lambda_q |q_i - q_j| \qquad i \in \mathcal{N}\backslash\mathcal{T}', j \in \mathcal{N}\backslash\mathcal{T} : i \neq j \tag{42}$$

Algorithm 2 presents the *shaw removal* operator, which is based on the procedure described in [17,35]. In this algorithm, parameter $d \geq 1$ is a deterministic parameter that introduces some randomness in selecting the order to remove. Hence, a high value of d favors selections close to the selected order. This value has been set to five based on the literature [52,53] and preliminary experiments.

Algorithm 2: Shaw removal

Data: (S, ρ, d)
Initialization Phase:
$removelist \leftarrow [\ ]$;
$removelist \leftarrow removelist \cup$ **Select_Random_Order**(S);
Iterative Phase:
while $|removelist| < \lceil \rho S \rceil$ **do**
  $j \leftarrow$ **Select_Random_Order**$(removelist)$;
  for *all orders* $i \in S \,|\, i \notin removelist$ **do**
    $L \leftarrow L + \{$**Calculate_Similarity_Score**$(i, j)\}$;
  Sort L such that $L_i \le L_{i+1}\ \ \forall i \in L$;
  $r \leftarrow \text{rand}(0, 1)$;
  $o \leftarrow L_{\lfloor |L| \times r^d \rfloor}$;
  $removelist \leftarrow removelist \cup o$;
Remove: orders in $removelist$ from S

- **Least paid removal**: In the least paid removal, the order with the lowest revenue p_i is removed from the current solution. The logic behind this operator is that the order generating the least revenue is the best order to remove and switch with another order that can generate more revenue.
- **Least profit removal** (worst removal): This removal destroys the solution in a similar way to the least paid removal operator. Instead of the order with the lowest revenue p_i, the order with the lowest profit is removed. In this case, the profit is calculated as the difference between the revenue p_i and all costs that can be attributed to each order, i.e., (*i*) travel costs, (*ii*) loading and unloading costs, (*iii*) waiting costs (if any) for early arrivals, and (*iv*) lateness costs. The reasoning behind this operator is that orders with a low revenue are not necessarily bad orders to include in the solution if the additional costs of including them are relatively low.
- **Longest time removal**: This time-based destroy operator selects the orders that take the highest times to serve. Serving times, in this case, are calculated as the summation of the travel times from (*i*) the location before the pickup location to the pickup location and (*ii*) the location before the delivery location to the delivery location plus the service times at both locations used for (un)loading the order. The idea behind this operator is that orders taking a long time are probably not convenient orders and, therefore, can be replaced by other less time-consuming orders.

4.4. Repair operators

Three repair operators are utilized in this ALNS metaheuristic. The following repair operators have been defined based on [17,35,39], and [51]:

- **Random insertion**: This repair operator iterates over each vehicle and selects a random unserved order that is not in the rejected list of the vehicles. Then, the pickup location of this order is added at a random location. The delivery location of this order is also randomly placed at a location after the pickup location. This process is repeated until all the insertion positions have been assessed. After this, if the order can still not be added, the order is deleted from the route and added to the rejected list of that vehicle. To avoid large computation times, the process is repeated until the number of orders in each rejected-order list equals r.
- **Basic greedy insertion**: This procedure assesses the closest pickup location of unserved orders, which are not in the rejected list of the vehicles. Hence, per vehicle, one of the orders is randomly selected, and the pickup location is added to the route. The procedure determines the best location for the delivery location of that order in the route. If all the positions have been evaluated and no feasible solution has been found, the order is erased from the route and added to the vehicle's rejected-orders list. This process is repeated for all other vehicles until the number of orders in each rejected-order list is at least equal to r.
- **Regret-based insertion**: For each unserved order that is not in the rejected list of that vehicle, the regret-based insertion operator inserts the order with the highest regret value. This regret value is the difference between the best insertion (of both the pickup and delivery location) and the current objective. Here, the best insertion place must comply with the capacity, weight, and time-window constraints. The order with the highest regret value is then inserted in the best place for the objective function. If an order cannot be inserted (e.g., it has no best insertion place, and thereby, the regret value is zero), it is added to the vehicle's rejected orders list. The regret-based insertion tries to deal with the insertion of more difficult orders, where one or multiple orders have already been incorporated into the routes. The process of adding orders with the highest regret factor is repeated until the number of rejected orders is equal to r.

4.5. Feasibility check

In this ALNS implementation, each time a new solution is constructed, a set of feasibility checks is applied. Algorithm 3 outlines the feasibility-check procedure for the SMDPDPMTWPD. The algorithm begins by setting a default value for the boolean feasibility parameter (line 1). Subsequently, for each vehicle and order (lines 2–11), the following conditions are verified. First, the load and weight must be non-negative and not exceed the maximum capacities Q^k and H^k (lines 4–5). Second, if the SMDPDPMTWPD extension includes soft time-window constraints, the maximum lateness is capped at 840 min to prevent trucks from arriving more than one day late (lines 6–8). Third, for case with hard-time windows, solutions are deemed infeasible if the arrival time is earlier than the start of the time window or if the departure time exceeds the window's closing time (lines 9–11). Finally, the boolean parameter $feasibility$ is returned (line 12).

Algorithm 3: Feasibility check

Input: Current solution x
Output: Feasibility status (true/false)
1 $feasibility \leftarrow$ true;
2 **for** *each* $k \in \mathcal{K}$ **do**
3   **for** *each* $i \in \mathcal{N}$ **do**
4   **if** $q_i^k < 0$ ***or*** $q_i^k > Q^k$ ***or*** $h_i^k < 0$ ***or*** $h_i^k > H^k$ **then**
5       $feasibility \leftarrow$ false;
6   **if** *Problem Version == Soft Time Windows* **then**
7   **if** $m_i > 840$ **then**
8         $feasibility \leftarrow$ false;
9   **else**
10   **if** $a_i^k < \sum_{t \in \mathcal{W}_i} b_i^t e_i^t$ ***or*** $d_i^k > \sum_{t \in \mathcal{W}_i} b_i^t l_i^t$ **then**
11         $feasibility \leftarrow$ false;
12 **return** $feasibility$;

5. Results and discussion

This section presents the results and discussion on the numerical experiments. A detailed description of the computational settings and parameter tuning is presented in Section 5.1. Then, Section 5.2 discusses the main assumptions and simplifications needed for the experiments. Section 5.3 describes the instance generation for the experiments. Section 5.4 outlines the main purpose of every set of experiments. The numerical results are shown in Section 5.5 - Section 5.10. Finally, the managerial insights and discussion of the results are provided in Section 5.11.

Table 6
Parameter values utilized in the calibration of ALNS and SA.

Parameter	SA	ALNS
r	$\{10, 15, 20\}$	*fixed from SA*
N_{iter}	$\{10, 15, 20\}$	*fixed from SA*
α	$\{0.95, 0.97, 0.99\}$	*fixed from SA*
T_{start}	$\{400, 500, 600\}$	$\{100, 200, 400\}$
T_{end}	1	1
ρ	–	$\{0.5, 0.7, 0.9\}$
η	–	$\{400, 500, 600\}$
$\sigma_1, \sigma_2, \sigma_3$	–	$\{(5, 1, 3), (1, 3, 5), (5, 3, 1)\}$
δ_T	30 min	30 min

5.1. Computational settings and parameter tuning

The MILP formulations and metaheuristics were implemented in Python 3.12.2. The MILP model was solved using the Gurobi 12.0.0 optimization software. All experiments were conducted on a computer with a 2.29 GHz Intel Xeon Gold 5218 CPU, 64 GB of RAM, running Windows 10 (64-bit). For the metaheuristics, each instance is executed 10 times and the average results are reported.

In the experiments, we compare two metaheuristic approaches, i.e., the ALNS introduced in Section 4 and the Simulated Annealing (SA) heuristic presented in Appendix A. We applied the Friedman non-parametric test [54] to detect statistical differences between different combinations of parameters. In case of statistical differences, we applied the Nemenyi Post-hoc test [55] to identify which specific parameter combinations differ from each other and select the parameters with the best performance. As ALNS and SA have similar parameters, we first calibrated the parameters of SA and then fixed those parameters in the calibration of ALNS. Hence, we can compare the performance of these metaheuristics under fair scenarios, while also reducing the computational times during calibration. Additionally, to further reduce computational time, we fixed the final temperature T_{end} at 1 for both SA and ALNS, as preliminary experiments showed that variations in T_{end} had no significant impact on the performance of these metaheuristic algorithms. Table 6 presents the parameters and their values used for calibrating ALNS and SA, where a dash indicates that the parameter is not applicable to the algorithm. After performing the Friedman non-parametric test and Nemenyi Post-hoc test, the best combination of parameters for SA is as follows: $N_{iter} = 20$, $T_{start} = 600$, $\alpha = 0.5$, and $r = 15$. Similarly, the best combination of parameters for ALNS is $N_{iter} = 20$, $T_{start} = 200$, $\alpha = 0.5$, $r = 15$, $\rho = 0.5$, $\eta = 0.2$, and $(\sigma_1, \sigma_2, \sigma_3) = (1, 3, 5)$.

5.2. Assumptions and simplifications

Several assumptions and simplifications have been made and considered while performing numerical experiments:

- All order data is complete, deterministic, and consistent with the real case under study.
- Travel distances are assumed to be symmetric and based on geodesic distances (e.g., the shortest path between two points along the surface of the Earth).[1] In practice, however, vehicles have to use roads to travel from one location to another that typically are not geodesic distances and, consequently, the real distance can be higher. Boyacı et al. [56] compares Euclidean distances in a VRP versus real-time road distances and concludes that road distances are typically approximately 30% higher than Euclidean distances (which varies per country and area). Although Euclidean distances are not used in this research, this 30% addition is considered.
- Travel times between two locations are based on the travel distance between the two locations divided by the average speed of trucks and rounded to minutes. We assume an average truck speed of $v = 80$ km/h for trucks in Europe [57]. According to Rietveld et al. [58], mean reported travel time is approximately 24% higher than network times given by route planners due to (1) non-driving time components (among others, traffic jams), (2) an underestimation of travel times by route planners and (3) the decision of drivers to use other routes. To incorporate both longer travel times than reported and legally allowed driving times, an additional increase of 40% is included in this calculation.
- To incorporate and consider night rest and working hours, a custom clock and time index is being used. Instead of a day from 0:00 till 23:59 with 24 h, a day from 6:00 till 19:59 with 14 h is used. The other hours are not incorporated as the driver is not allowed to work. If there is a time window (partially) overlapping the period of the day that is reserved for night rest (from 20:00 till 5:59), the time window is shortened for those cases hard time windows are addressed (thus, it starts at 6:00 or ends at 19:59) or fully omitted in case the complete time window is in the night rest.
- For (un)loading a vehicle, a fixed service time of 120 min is assumed if the size of the freight is at least half of a truck (i.e., 6.8 LDMs) and 60 min otherwise.
- The cost factors are defined as follows. The routing costs per kilometer are assumed to be $c_{ij} = €\ 0.86$. The cost factor per unit of time for the service time is set to be equal to the employee cost per minute: $f^t = \frac{25}{60} \approx €\ 0.40$. Service can immediately start when arriving at a pickup or delivery location and has been set to $c_i = €0.10$. These cost factors are in line with values that have been found in the literature [59,60]. The emission market price has been set to $c_e = €\ 0.08421$ and is based on the average of the European Union's Emissions Trading Scheme allowance prices in the period January 2022 to September 2023.
- A conversion factor of $\alpha = 0.0854$ kg CO_2 per kilometer and loading meter has been used to calculate the CO_2 emissions of the vehicles.

5.3. Instance generation

The data needed for this research can be categorized into two classes: vehicle data and order data. Vehicle data consists of the number of vehicles for which freight is searched and per vehicle the name (for identification), the capacity (the space left and for which freight needs to be found in LDMs), the current weight, the maximum allowed weight, the start location (country and zip code) the time at which the vehicle becomes empty (start date and time), one or multiple ending locations (country and zip codes) and time at which the vehicle should be at one of its end locations (ending date and time). Three different numbers for vehicles are considered ($K = 1, 2$ or 4), where the parameters per vehicle are depicted in Table 7. These values are based on real-case scenarios of a transportation company with its headquarter in the east of the Netherlands. Note that the starting date of the model has been set to 05-02-2024.

Order data consists of the number of orders from which a selection needs to be made, the identification number of each order, pickup and delivery dates, number of consecutive days (with respect to the pickup or delivery date) during which pickup/delivery is (also) allowed, pickup and delivery locations (country and zip code) and time window(s) (identical in case of multiple pickup/delivery days), the number of kilometers between the pickup and delivery location, the revenue earned when the order is selected, the demand (in LDMs) and weight of the order and whether the offering company requires the exchange of pallets and other auxiliary transport materials. Four different numbers of orders are used, with values $n = 25, 50, 100$ or 250. Given the number of vehicles and orders, there are $3 \times 4 = 12$

[1] See https://geopy.readthedocs.io/en/stable/ for more information.

Table 7
Data for the vehicles used in the experiments.

Category	Attribute	Vehicle 1	Vehicle 2	Vehicle 3	Vehicle 4
General info	Capacity	13.6	10	13.6	6.0
	Weight	0	10 000	0	6500
	Max weight	24 000	24 000	24 000	24 000
Start location	Country code	DE	NL	DE	DE
	Zipcode	70 173	7547	20 038	80 331
	Date	05-02-2024	06-02-2024	06-02-2024	07-02-2024
	Time	6:00	6:00	9:00	11:00
End location(s)	Country code(s)	NL, NL, NL	DE	NL	NL, DE
	Zip code(s)	7547, 5656, 8021	10 115	7547	3454, 48 079
	Date	07-02-2024	09-02-2024	07-02-2024	09-02-2024
	Time	12:00	18:00	19:00	15:00

Table 8
Overview of the computational experiments and their goals.

Section	Problem variant				Goal
	HTW	HTW + CO_2	STW	STW + CO_2	
5.5	✓	✓	✓	✓	Compare the performance of ALNS with a Simulated Annealing metaheuristic framework.
5.6	✓				Assess the complexity of the problem and compare the results of the MILP with the best metaheuristic from Section 5.5.
5.7	✓				Test the ALNS algorithm and evaluate the transportation metrics for the profit maximization SMDPDPMTWPD.
5.8		✓			Assess the influence of CO_2 emissions in the objective function and the routing schedules.
5.9			✓		Determine the influence of soft time windows in the SMDPDPMTWPD and the routing schedules.
5.10				✓	Evaluate the influence of soft time windows and CO_2 in the routing schedules.

data instances. Each instance is labeled as D-X-Y, where D denotes the data instance, X represents the number of vehicles, and Y indicates the number of orders. Notably, instances with the same number of orders share the same set of orders (e.g., D-1-25, D-2-25, and D-4-25 use identical orders). However, when the number of orders increases, a new set of orders is generated (e.g., the orders in D-1-25, D-1-50, D-4-100, and D-4-250 are distinct).

Additionally to the real-case instances, we also adapted instances from the literature for validation. We took 16 instances from [5], as the problem studied in that paper resembles the SMDPDPMTWPD. The instances in [5] are labeled as *SFTa-bc-d-e*, where: *a* is the instance number, *b* depicts the instance generation method (R, C and RC), *c* defines the number of orders in the order pool, *d* represents the number of trucks, and *e* denotes the number of orders. In contrast to El Bouyahyiouy and Bellabdaoui [5], the SMDPDPMTWPD studied in this paper allows trucks to travel directly from the starting depot to the ending depot when no positive revenue can be obtained from picking up and delivering orders. Consequently, we replicated their approach to repair and adapt the instances for solving the SMDPDPMTWPD. The newly adapted instance set is available at https://github.com/Supernova2786/SMDPDPMTWPD_instances.

5.4. Experimental design

Table 8 provides an overview of the conducted experiments, including a description of their goal. First, we compare the performance of ALNS with a Simulated Annealing metaheuristic for the four studied problem variants (Section 5.5). Then, we compare the best metaheuristic from Section 5.5 with the MILP formulation on a small instance set to validate its performance and study the problem complexity (Section 5.6). After that, we provide managerial insights into the transportation metrics of vehicles when solving the four studied problem variants (Sections 5.7–5.10).

5.5. Comparison of metaheuristic approaches

This set of experiments aims to validate the performance of ALNS. Hence, we compare the ALNS proposed in Section 4.1 with the Simulated Annealing (SA) metaheuristic framework presented in Appendix A, for the profit maximization SMDPDPMTWPD. Each instance is executed 10 times, and the average results for both ALNS and SA are reported. Table 9 lists the average objective value (Z), standard deviation (Z_σ), percentage of improvement over the initial solution (ΔZ_{Init}), and computational time in seconds for the profit maximization variant (HTW). The best average objective values are bold-faced. Further details on this comparison for other problem variants are presented in Appendix B.

Results show that the ALNS metaheuristic significantly outperforms the average objective values found by SA, with €1322.49 for ALNS compared to €605.54 for SA, respectively. Hence, ALNS outperforms SA in 25 out of the 28 tested instances. In this context, the standard deviation values of ALNS were 10 times smaller than the SA, showing a more stable performance of ALNS. Furthermore, the results show that the percentage improvement of the objective function values over the initial solutions was on average 308.17 for ALNS and 223.78 for the SA. This demonstrates that ALNS was able to explore the solution space more effectively than SA for this set of instances, but at the expense of higher computational times. Appendix B shows the same comparison for the other problem variants studied in this paper, providing similar results. Consequently, we utilize the ALNS metaheuristic to analyze the managerial insights in the subsequent experiments and further disregard the SA.

5.6. Performance evaluation of ALNS and the MILP model

This set of experiments compares the best metaheuristic algorithm from Section 5.5, i.e., ALNS, with the MILP model. Table 10 presents the results of both the MILP formulation and ALNS metaheuristic, tested

Table 9
Average results of ALNS and SA metaheuristic for the profit maximization SMDPDPMTWPD.

Exp.	Data instance	ALNS				SA			
		Z (€)	Z_σ	ΔZ_{Init} (%)	Time (s)	Z (€)	Z_σ	ΔZ_{Init} (%)	Time (s)
Profit maximization	D-1-25	**979.05**	304.55	154.39	112.69	726.33	626.99	99.37	21.91
	D-1-50	**2830.15**	87.65	423.41	69.68	622.68	1538.38	48.93	10.49
	D-1-100	**4268.95**	52.89	850.03	507.58	2630.87	1633.80	645.25	29.58
	D-1-250	**2602.08**	821.66	427.68	637.18	1335.83	1176.66	189.42	20.00
	D-2-25	256.82	290.21	149.82	71.35	**2831.63**	1923.24	462.13	79.32
	D-2-50	2142.69	93.80	269.38	88.82	**2278.11**	1904.11	513.74	18.11
	D-2-100	**3573.12**	52.89	1993.35	280.68	1408.51	2056.85	294.20	72.96
	D-2-250	**5997.39**	40.58	1478.04	1811.42	3497.93	1702.46	491.19	107.57
	D-4-25	92.22	151.07	113.90	119.15	**4493.18**	1866.07	1732.85	86.86
	D-4-50	**1302.13**	158.97	147.52	210.53	309.87	1410.17	430.59	33.62
	D-4-100	**2746.90**	135.70	216.44	583.18	1348.78	2278.79	156.30	98.54
	D-4-250	**5643.23**	192.66	284.89	1800.19	4524.75	2880.55	235.40	145.88
	SFT1-C25-16-2	**−156.88**	10.95	75.53	99.86	−497.78	128.99	28.36	214.39
	SFT2-C25-16-2	**−470.83**	0.00	45.43	94.76	−657.77	104.24	26.12	194.21
	SFT1-C50-24-3	**773.92**	0.00	206.96	391.45	−145.27	423.00	80.71	214.73
	SFT2-C50-24-3	**127.62**	13.51	114.29	238.35	−364.96	386.45	71.13	187.91
	SFT1-R25-20-2	**396.63**	29.09	171.91	234.08	−233.56	308.64	137.57	102.02
	SFT2-R25-20-2	**−97.67**	0.00	86.00	281.90	−525.10	129.90	27.08	107.10
	SFT1-R50-30-3	**342.75**	0.00	144.03	464.49	−427.20	264.55	60.44	141.62
	SFT2-R50-30-3	**−407.92**	0.00	66.26	189.05	−766.97	88.07	39.08	153.01
	SFT1-RC25-20-2	**106.08**	9.78	114.43	250.82	−346.82	370.43	72.22	112.43
	SFT2-RC25-20-2	**107.42**	0.00	118.49	93.45	−389.58	217.77	53.01	108.92
	SFT1-RC50-30-3	**614.00**	0.00	165.88	575.42	−348.42	270.37	49.76	135.37
	SFT2-RC50-30-3	**1286.50**	0.00	313.28	558.48	316.18	689.57	155.32	145.62
	SFT1-R100-50-5	**50.98**	8.37	102.98	935.13	−978.06	215.02	35.82	269.83
	SFT2-R100-50-5	**206.72**	0.42	111.31	862.80	−1391.87	310.63	37.52	247.29
	SFT1-R100-75-7	**910.60**	11.84	142.27	1814.59	−1129.03	573.24	49.49	353.46
	SFT2-R100-75-7	**804.97**	92.41	140.80	1816.05	−1167.18	352.93	42.74	334.21
	Average	**1322.49**	91.39	308.17	542.61	605.54	922.57	223.78	133.82

on a small set of instances involving one and two vehicles (K), and between five to twenty orders (n). The columns report the best solution found Z (€), the optimality Gap (%), the percentage difference between ALNS and MILP solutions Δ_{MILP} (%), and computational times (sec). The percentage difference is computed as $\Delta_{MILP} = \frac{Z_{MILP}-Z_{ALNS}}{Z_{MILP}} \cdot 100$, where Z_{MILP} and Z_{ALNS} represent the best solutions of the MILP formulation and ALNS metaheuristic, respectively. Further details on these computational experiments (spread in objective function values) can be found in Appendix C.

The results show that increasing the number of vehicles may lead to negative objective values, i.e., financial losses. For the instance $n = 5$ and $K = 1$, the scenario fails to produce revenue, revealing an optimal value of €−617.75. When the number of vehicles increases to 2, the optimal solution shows values of €−1313.58 for the instance $n = 5$ and $K = 2$. The same tendency is observed for the instance $n = 10$, where the optimal results decrease from €609.75 to €−86.08 for the scenarios with $K = 1$ and $K = 2$ vehicles, respectively. The average values for the optimal solutions decrease from €575.63 to €−120.21 when the number of vehicles increases from $K = 1$ to $K = 2$. Furthermore, the results also show that the freight exchange platforms offer a significant potential for generating income. Despite the instances where $n = 5$ and $n = 10$ exhibit negative objective values, instances where $n \geq 15$ result in positive objective values. This shows that freight exchange platforms are a successful mechanism for reducing empty kilometers of return trips and can generate additional income.

The results validate the performance of ALNS for the small instances. Hence, for all the tested instances, ALNS finds the same optimal solutions as the MILP formulation, showing an average percentage difference (Δ_{MILP}) of 0.00% for the instances with one and two vehicles. This is achieved in significantly smaller computational times, where ALNS and MILP report average values of 57.93 and 241.93 (s) for instances with one vehicle, respectively, and average values of 41.43 and 343.91 (s) for the instances with two vehicles. This shows that ALNS finds the same optimal solutions as the MILP in significantly smaller computational times, validating the effectiveness of our metaheuristic approach. Furthermore, results also show how the complexity of the problem increases with larger numbers of vehicles, as ALNS and MILP report larger computational times when the number of vehicles increases from 1 to 2. Similar results are observed with larger number of orders. Additionally, the MILP formulation presents optimality gaps larger than 1000% for instances with larger numbers of orders ($n \geq 20$, not shown in the tables). This shows that the complexity of the problem significantly increases with larger numbers of orders. Therefore, we conclude that the ALNS metaheuristic is well-suited for addressing large instances.

5.7. Analysis on the profit maximization SMDPDPMTWPD

Table 11 presents the results of the ALNS metaheuristic for the profit maximization SMDPDPMTWPD, tested on a large set of instances. The columns report the average objective value Z (€) across the 10 executions of each instance, the initial solution Z_{Init} (€), improvement percentage (ΔZ_{Init}) of Z compared to the initial solution, computational times (s), total waiting time for trucks (in minutes), total delay time for trucks (in minutes), total traveled distance (in kilometers), total travel time (in minutes), number of selected orders (Nr. of orders), percentage of the selected orders that occupy the full truckload (orders with 13.6 LDMs), and the percentage of the travel time that a truck is loaded up to 80% or more than of its capacity. The percentage improvement is computed as $\Delta Z_{Init} = \frac{Z-Z_{Init}}{|Z_{Init}|} \cdot 100$. Further information on these computational experiments (spread in values) is presented in Appendix D.

Results show that ALNS effectively explores the solution space. In particular, ALNS reports average objective values of €1322.49, whereas the initial solutions (Z_{Init}) present average objective values of €−1228.83. ALNS achieves this with average computational times of 542.61 s. Consequently, ALNS outperforms the initial solutions by 308.17%, showing that our metaheuristic algorithm effectively explores

Table 10
Comparison of the ALNS metaheuristic with the MILP model on a small set of instances. Note that the results of ALNS are the maximum values seen across 10 replications for fair comparison with the MILP.

n	$K = 1$						$K = 2$					
	MILP			ALNS			MILP			ALNS		
	Z (€)	Gap (%)	Time (s)	Z (€)	Δ_{MILP} (%)	Time (s)	Z (€)	Gap (%)	Time (s)	Z (€)	Δ_{MILP} (%)	Time (s)
5	−617.75	0.00	0.24	−617.75	0.00	4.65	−1313.58	0.00	0.28	−1313.58	0.00	8.64
10	609.75	0.00	23.18	609.75	0.00	26.54	−86.08	0.00	33.57	−86.08	0.00	35.14
15	1155.25	0.00	65.14	1155.25	0.00	89.60	459.42	0.00	121.82	459.42	0.00	48.61
20	1155.25	0.00	879.16	1155.25	0.00	110.93	459.42	0.00	1219.96	459.42	0.00	73.31
Avg	575.63	0.00	241.93	575.63	0.00	57.93	−120.21	0.00	343.91	−120.21	0.00	41.43

Table 11
Results of the ALNS metaheuristic for the profit maximization SMDPDPMTWPD (HTW).

Exp.	Data instance	Z (€)	Z_{Init} (€)	ΔZ_{Init} (%)	Time (s)	Waiting (min)	Late (min)	Distance (km)	Travel time (min)	Nr. of orders	% FTL orders	% Travel time ≥80% loaded
Profit maximization	D-1-25	979.05	−203.12	154.39	112.69	62.60	0.00	1635.93	1500.40	1.80	60.00	26.28
	D-1-50	2830.15	316.01	423.41	69.68	17.00	0.00	1298.95	1184.80	1.00	100.00	49.74
	D-1-100	4268.95	−504.78	850.03	507.58	17.00	0.00	1512.56	1428.40	2.00	100.00	63.11
	D-1-250	2602.08	−634.74	427.68	637.18	109.10	0.00	1517.67	1369.60	1.40	85.00	40.97
	D-2-25	256.82	−1185.07	149.82	71.35	62.60	0.00	2239.30	1988.40	1.80	60.00	19.69
	D-2-50	2142.69	−1042.89	269.38	88.82	17.00	0.00	1873.37	1649.10	1.00	100.00	35.70
	D-2-100	3573.12	−693.59	1993.35	280.68	17.00	0.00	2093.95	1898.40	2.00	100.00	47.48
	D-2-250	5997.39	−1957.48	1478.04	1811.42	401.00	0.00	4226.05	3852.30	4.00	25.00	30.24
	D-4-25	92.22	−1543.20	113.90	119.15	229.00	0.00	3395.23	2943.50	2.00	100.00	29.66
	D-4-50	1302.13	−2899.57	147.52	210.53	163.00	0.00	3264.19	2824.20	2.00	100.00	28.97
	D-4-100	2746.90	−2733.58	216.44	583.18	163.00	0.00	3472.79	3063.80	3.00	100.00	36.91
	D-4-250	5643.23	−3389.09	284.89	1800.19	481.80	0.00	5603.60	5003.80	4.40	46.00	29.82
	SFT1-C25-16-2	−156.88	−688.70	75.53	99.86	33.30	0.00	400.93	324.50	2.00	100.00	23.13
	SFT2-C25-16-2	−470.83	−867.16	45.43	94.76	9.00	0.00	420.93	341.00	1.00	100.00	8.80
	SFT1-C50-24-3	773.92	−848.97	206.96	391.45	8.00	0.00	975.58	789.00	6.00	100.00	42.59
	SFT2-C50-24-3	127.62	−945.84	114.29	238.35	14.20	0.00	798.26	643.60	4.40	100.00	30.71
	SFT1-R25-20-2	396.63	−673.83	171.91	234.08	29.00	0.00	816.51	657.80	4.00	100.00	35.61
	SFT2-R25-20-2	−97.67	−722.59	86.00	281.90	31.00	0.00	650.00	525.00	4.00	100.00	36.19
	SFT1-R50-30-3	342.75	−879.93	144.03	464.49	45.00	0.00	795.35	642.00	5.00	100.00	51.87
	SFT2-R50-30-3	−407.92	−1216.03	66.26	189.05	42.00	0.00	461.63	373.00	2.00	100.00	30.03
	SFT1-RC25-20-2	106.08	−759.72	114.43	250.82	93.50	0.00	923.02	746.80	4.00	75.00	45.66
	SFT2-RC25-20-2	107.42	−654.57	118.49	93.45	38.00	0.00	486.05	393.00	2.00	50.00	18.07
	SFT1-RC50-30-3	614.00	−958.58	165.88	575.42	0.00	0.00	1251.16	1008.00	6.00	100.00	56.35
	SFT2-RC50-30-3	1286.50	−899.96	313.28	558.48	35.00	0.00	1094.19	883.00	6.00	83.33	67.61
	SFT1-R100-50-5	50.98	−1757.43	102.98	935.13	111.40	0.00	1440.58	1164.00	7.90	74.64	42.05
	SFT2-R100-50-5	206.72	−1843.42	111.31	862.80	71.00	0.00	1387.44	1122.00	6.00	100.00	33.69
	SFT1-R100-75-7	910.60	−2223.03	142.27	1814.59	88.10	0.00	2183.60	1762.30	13.70	92.69	46.60
	SFT2-R100-75-7	804.97	−1996.33	140.80	1816.05	37.50	0.00	2132.79	1720.10	12.70	92.09	47.29
	Average	1322.49	−1228.83	308.17	542.61	86.65	0.00	1726.84	1492.92	4.04	87.28	37.67

the solution space when optimizing the non-compulsory pickup of orders for return trips. Furthermore, the routing schedules show that the vehicles, on average, travel 1726.84 km, in 1492.92 min, and wait 86.65 min to serve 4.04 orders. Therefore, we conclude that the freight exchange platforms represent an effective approach to dealing with empty return trips. Thus, the vehicles only needed to serve 4.04 orders, on average, to generate profits. This highlights the positive impact of freight exchange platforms and also that a small number of orders can mitigate the costs of empty return trips.

5.8. Analysis on the profit maximization including CO_2 emissions SMDPDPMTWPD

Table 12 presents the results of the ALNS metaheuristic for the profit maximization SMDPDPMTWPD with CO_2 emissions costs, tested on a large set of instances. The columns list the same information as the columns in Table 11. Further information on these computational experiments is presented in Appendix D.

The results show that the ALNS metaheuristic averages objective values of €1267.03, whereas the initial solutions (Z_{Init}) report average objective values of €−1281.73. These results validate the performance of the ALNS metaheuristic, which outperforms the average values of the initial solutions by 641.39%, within computational times of 559.41 s. Overall, we observe a slight decrease in the average objective values for the SMDPDPMTWPD solutions when incorporating CO_2 emissions costs. Nevertheless, this decrease does not affect the performance of the ALNS metaheuristic nor significantly impact the routing schedules. In the results, the vehicles travel 1728.42 km, in 1494.98 min, and wait 87.74 min to serve 4.09 orders. Thus, we conclude that even though incorporating CO_2 emissions costs into the profit maximization can lead to lower incomes, those new sustainable solutions do not significantly compromise the business operations and large changes to the vehicle routing schedules. Therefore, the freight exchange platforms also show a strong potential to deal with empty kilometers of return trips when incorporating CO_2 emissions costs.

5.9. Analysis on the soft time windows SMDPDPMTWPD

Table 13 presents the results of the ALNS metaheuristic for the soft time windows SMDPDPMTWPD, tested on a large set of instances. The columns list the same information as the columns in Table 11. Further information on these computational experiments is presented in Appendix D.

The ALNS metaheuristic shows average objective values of €3696.93, whereas the initial solutions (Z_{Init}) report average values of €−3284.26. Compared to the results for the hard time-window

Table 12
Results of the ALNS metaheuristic for the profit maximization with CO_2 emissions costs SMDPDPMTWPD (HTW + CO_2).

Exp.	Data instance	Z (€)	Z_{Init} (€)	ΔZ_{Init} (%)	Time (s)	Waiting (min)	Late (min)	Distance (km)	Travel time (min)	Nr. of orders	% FTL orders	% Travel time ≥80% loaded
	D-1-25	912.26	−452.94	161.09	114.13	62.60	0.00	1650.58	1512.40	1.80	60.00	26.11
	D-1-50	2763.26	−526.86	322.88	67.44	17.00	0.00	1309.07	1192.90	1.00	100.00	49.43
	D-1-100	4188.22	300.87	552.67	478.79	17.00	0.00	1509.77	1426.20	2.00	100.00	63.22
	D-1-250	2503.01	−255.89	326.01	663.44	103.70	0.00	1577.09	1419.10	1.40	80.00	41.57
	D-2-25	282.47	−1194.69	159.56	79.28	44.30	0.00	2267.79	2018.70	1.90	55.00	18.36
	D-2-50	2064.63	−1715.75	263.57	92.16	17.00	0.00	1892.79	1664.80	1.00	100.00	35.36
	D-2-100	3497.12	−1362.26	445.19	300.11	17.00	0.00	2087.21	1893.00	2.00	100.00	47.62
	D-2-250	5889.08	−2163.96	499.88	1576.42	401.00	0.00	4223.37	3850.20	4.00	25.00	30.26
	D-4-25	27.80	−2240.50	101.71	128.54	229.00	0.00	3381.16	2932.00	2.00	100.00	29.76
	D-4-50	1223.24	−2395.47	227.73	224.22	163.00	0.00	3266.40	2826.10	2.00	100.00	28.93
	D-4-100	2567.32	−1965.62	227.92	608.01	154.20	0.00	3435.12	3029.00	2.90	95.00	36.66
	D-4-250	5552.79	−2778.24	12 409.16	1815.13	513.60	0.00	5622.79	5038.10	4.80	44.00	29.49
Profit maximization with CO_2 emissions	SFT1-C25-16-2	−161.45	−718.15	76.23	108.57	33.10	0.00	397.21	321.50	2.00	100.00	23.34
	SFT2-C25-16-2	−474.47	−855.62	44.31	101.37	9.00	0.00	420.93	341.00	1.00	100.00	8.80
	SFT1-C50-24-3	733.43	−912.79	198.95	434.29	8.00	0.00	975.58	789.00	6.00	100.00	42.59
	SFT2-C50-24-3	126.54	−922.62	114.26	270.91	25.70	0.00	788.14	635.20	4.80	100.00	32.92
	SFT1-R25-20-2	372.00	−815.36	148.13	258.78	26.00	0.00	823.02	663.00	4.20	100.00	36.42
	SFT2-R25-20-2	−120.53	−672.92	80.00	308.71	31.00	0.00	650.00	525.00	4.00	100.00	36.19
	SFT1-R50-30-3	302.49	−1303.77	123.64	506.83	45.00	0.00	795.35	642.00	5.00	100.00	51.87
	SFT2-R50-30-3	−421.56	−1180.52	63.74	218.05	42.00	0.00	461.63	373.00	2.00	100.00	30.03
	SFT1-RC25-20-2	63.29	−755.18	108.85	276.48	97.00	0.00	923.49	747.20	4.00	75.00	45.64
	SFT2-RC25-20-2	93.75	−819.18	112.94	103.54	38.00	0.00	486.05	393.00	2.00	50.00	18.07
	SFT1-RC50-30-3	544.97	−1030.73	157.97	643.69	0.00	0.00	1251.16	1008.00	6.00	100.00	56.35
	SFT2-RC50-30-3	1215.84	−790.52	553.75	624.38	35.00	0.00	1094.19	883.00	6.00	83.33	67.61
	SFT1-R100-50-5	−6.65	−1672.12	99.58	1058.85	112.50	0.00	1383.72	1118.20	7.90	74.45	41.54
	SFT2-R100-50-5	163.12	−2162.17	107.63	975.11	88.50	0.00	1379.88	1115.00	6.00	100.00	33.90
	SFT1-R100-75-7	804.84	−2158.94	138.84	1813.81	86.10	0.00	2199.53	1775.10	14.00	92.86	46.68
	SFT2-R100-75-7	769.91	−2366.58	132.75	1812.31	40.30	0.00	2142.67	1727.80	12.90	92.24	48.11
	Average	1267.03	−1281.73	641.39	559.41	87.74	0.00	1728.42	1494.98	4.09	86.67	37.74

SMDPDPMTWPD in Section 5.7, we observe a significant increase in the average objective values, when considering soft time windows. As expected, soft time windows enlarge the solution space of the SMDPDPMTWPD. Consequently, ALNS reports an improvement of 430.37% over the initial solution, with computational times averaging 1048.92 s. In the routing schedules, the vehicles travel 2304.57 km in 2012.58 min, wait 180.72 min to serve 5.83 orders, and experience 1140.45 min of delayed deliveries. Soft time windows significantly impact the routing schedules and the profitability, as we observe a reduction in travel and waiting times compared to the hard time window scenarios. Moreover, for the soft time window SMDPDPMTWPD, trucks increase the number of served orders to 5.83. This indicates that even though the delayed times average 1140.45 min, the related costs of delayed deliveries do not mitigate the profits of serving more orders. Interestingly, the results show that when allowing soft time windows, the percentage of FTL orders slightly decreased to 80.11%, whereas the travel time ≥ 80% loaded marginally increased to 39.73%, respectively. These results highlight that there is no strong relation between the occupation rates of the vehicles and the generated profits.

5.10. Analysis on the soft time windows including CO_2 emissions SMDPDPMTWPD

Table 14 presents the results of the ALNS metaheuristic for the soft time windows with CO_2 emissions costs SMDPDPMTWPD, tested on a large set of instances. The columns list the same information as the columns in Table 11. Further information on these computational experiments is presented in Appendix D.

The ALNS metaheuristic shows average objective values of €3636.08, whereas the initial solutions (Z_{Init}) report average values of €−3140.79. Compared to the results in Sections 5.7 and 5.8, we observe a similar tendency when incorporating CO_2 emissions costs into the soft time-window SMDPDPMTWPD: there is a slight decrease in the average objective values. Nevertheless, this decrease does not strongly affect the routing schedules of the SMDPDPMTWPD. ALNS reports average improvements of 2738.12% over the initial solution, within computational times of 1046.00 s. In particular, these results validate the importance of the soft time-window approach to exploit the benefits of the freight exchange platforms when solving SMDPDPMTWPD. Even though there is a slight decrease in the average objective values, the transportation metrics of the routing schedules barely change when including CO_2 emissions costs. In the results, the vehicles travel on average 2270.09 km in 1982.38 min, wait 179.45 min to serve 5.65 orders, and experience 1088.23 min of delayed deliveries. Thus, the vehicle occupation metrics show similar rates with average values of 79.14% and 39.01% for the % FTL orders and % Travel time ≥ 80% loaded, respectively.

5.11. Managerial insights and discussion

This section summarizes the managerial insights and presents a discussion across the conducted sets of experiments. Fig. 2 illustrates the average values of the ALNS algorithm when solving the following SMDPDPMTWPD variants: (hard time windows) maximizing profits (HTW), (hard time windows) maximizing profits with CO_2 emissions costs (HTW + CO_2), soft time windows maximizing profits (STW), and soft time windows maximizing profits with CO_2 emissions costs (STW + CO_2) information of the objective function values and number of orders for each problem variant. The average values correspond to the objective function values (Z), and number of served orders. Similarly, Table 15 summarizes the transportation metrics for the same SMDPDPMTWPD variant. The columns report the average values of truck waiting time (min), covered distance (km), and travel time (min).

The results show a slight decrease in the average objective values when including the CO_2 emissions costs. As expected, the profit is less than excluding the CO_2 emissions costs. Nevertheless, these additional costs slightly affect the routing schedules. Hence, when comparing the transportation metrics of HTW and HTW+CO_2 in Tables 11 and 12, respectively, the covered distances and travel times show the same results for the 28 studied instances.

Table 13
Results of the ALNS metaheuristic for the profit maximization SMDPDPMTWPD with soft time windows (STW).

Exp.	Data instance	Z (€)	Z_{Init} (€)	ΔZ_{Init} (%)	Time (s)	Waiting (min)	Late (min)	Distance (km)	Travel time (min)	Nr. of orders	% FTL orders	% Travel time ≥80% loaded
Soft time windows profit maximization	D-1-25	2 151.40	−472.93	734.96	288.01	209.00	267.10	1659.42	1540.10	2.00	100.00	56.70
	D-1-50	4 374.31	−456.82	477.09	553.65	35.20	855.00	2324.19	2064.30	2.00	65.00	28.42
	D-1-100	5 713.38	−111.47	2172.67	1580.92	17.60	1826.80	2242.33	2066.90	2.80	95.00	51.74
	D-1-250	11 400.91	432.88	1108.71	1830.02	0.40	1552.10	2082.79	1888.60	2.50	97.50	46.55
	D-2-25	2 332.36	−1985.37	365.11	766.82	819.00	1772.70	3586.05	3266.70	6.00	33.33	26.70
	D-2-50	4 377.11	−2324.73	508.76	756.82	290.80	960.40	3564.77	3176.60	3.00	40.00	17.64
	D-2-100	5 916.66	−2251.47	496.84	1785.17	238.00	2024.90	3444.88	3146.60	3.90	74.17	35.68
	D-2-250	15 056.69	−1937.93	1686.16	1850.71	331.00	2537.70	4173.60	3679.50	6.30	36.19	44.08
	D-4-25	11 697.05	−3574.20	615.42	497.03	1394.10	2791.10	5645.58	5038.00	6.90	43.57	27.12
	D-4-50	5 848.25	−3618.41	371.90	559.77	339.70	2104.00	5785.23	5135.30	4.30	58.00	24.85
	D-4-100	8 322.08	−5778.85	256.43	1383.13	238.00	3698.00	5803.95	5209.80	5.40	81.33	34.75
	D-4-250	21 007.81	−3681.07	1514.70	1847.80	373.40	6604.10	6902.09	6158.40	10.10	43.62	39.76
	SFT1-C25-16-2	−163.68	−2059.42	92.02	93.22	33.60	0.00	406.51	329.00	2.00	100.00	22.81
	SFT2-C25-16-2	−470.83	−2432.27	80.55	71.96	9.00	0.00	420.93	341.00	1.00	100.00	8.80
	SFT1-C50-24-3	823.10	−2411.45	134.52	1257.33	76.00	109.00	1047.67	848.00	8.00	100.00	49.41
	SFT2-C50-24-3	137.72	−2775.67	105.03	341.04	23.00	67.50	803.37	647.40	4.90	100.00	32.62
	SFT1-R25-20-2	496.36	−2181.32	123.23	733.93	41.00	379.20	696.63	562.90	4.00	100.00	41.65
	SFT2-R25-20-2	−97.67	−3058.59	96.80	467.15	31.00	0.00	650.00	525.00	4.00	100.00	36.19
	SFT1-R50-30-3	348.32	−4039.08	108.68	998.98	28.00	21.00	884.88	715.00	6.00	100.00	52.31
	SFT2-R50-30-3	−407.92	−4641.04	91.16	281.96	42.00	0.00	461.63	373.00	2.00	100.00	30.03
	SFT1-RC25-20-2	218.73	−2331.43	109.52	700.81	28.00	786.00	786.05	636.00	4.00	75.00	53.62
	SFT2-RC25-20-2	115.42	−2560.61	104.58	124.39	38.00	2.00	489.30	395.40	2.40	56.67	19.67
	SFT1-RC50-30-3	760.70	−3301.58	123.30	1811.56	20.00	654.30	1452.09	1170.10	9.80	100.00	70.50
	SFT2-RC50-30-3	1 321.22	−2917.40	145.89	1629.35	35.00	499.70	1381.51	1116.30	7.90	87.32	71.37
	SFT1-R100-50-5	124.27	−6647.26	101.87	1821.86	139.50	429.80	1624.19	1311.90	11.00	81.76	49.75
	SFT2-R100-50-5	205.19	−7117.25	102.91	1624.80	69.40	245.90	1427.56	1154.40	7.00	100.00	36.56
	SFT1-R100-75-7	930.67	−9428.40	109.89	1854.96	100.60	657.60	2237.21	1804.20	16.80	85.15	48.43
	SFT2-R100-75-7	974.40	−8296.17	111.78	1856.67	59.90	1086.70	2543.49	2051.70	17.20	89.57	54.61
	Average	3 696.93	−3284.26	430.37	1048.92	180.72	1140.45	2304.57	2012.58	5.83	80.11	39.73

Table 14
Results of the ALNS metaheuristic for the profit maximization with CO_2 emissions costs and soft time windows SMDPDPMTWPD (STW + CO_2).

Exp.	Data instance	Z (€)	Z_{Init} (€)	ΔZ_{Init} (%)	Time (s)	Waiting (min)	Late (min)	Distance (km)	Travel time (min)	Nr. of orders	% FTL orders	% Travel time ≥80% loaded
Soft time windows profit maximization with CO_2 emissions	D-1-25	2 103.84	44.18	216.29	290.89	209.00	245.70	1633.02	1518.70	2.00	100.00	57.46
	D-1-50	4 365.57	−713.81	630.03	579.93	0.00	942.20	2390.93	2112.70	2.00	65.00	26.76
	D-1-100	5 293.50	−69.07	839.20	1382.85	35.20	1345.80	2129.30	1945.80	2.40	90.00	46.63
	D-1-250	10 866.02	−190.35	1 002.91	1822.73	0.10	1573.70	2137.56	1938.10	2.40	93.33	45.07
	D-2-25	2 170.93	−732.08	872.63	783.08	819.00	1794.10	3612.44	3288.10	6.00	33.33	26.53
	D-2-50	4 122.57	−1364.37	371.30	769.88	255.60	943.80	3494.07	3108.20	3.00	46.67	18.93
	D-2-100	5 894.24	−1258.48	1 335.15	1817.71	238.00	2075.60	3448.60	3159.60	4.00	75.00	36.99
	D-2-250	15 068.48	−1934.77	1 579.17	1876.24	331.00	2628.80	4212.33	3715.20	6.40	37.14	44.14
	D-4-25	11 522.68	−3087.20	506.83	498.75	1393.60	2720.30	5611.28	5044.40	6.80	44.57	27.12
	D-4-50	6 032.62	−3775.13	65 031.87	565.61	271.90	2144.50	5893.60	5200.30	4.10	53.50	21.88
	D-4-100	7 853.49	−3498.85	1 912.80	1473.26	294.10	3358.90	5790.47	5194.80	5.40	73.33	32.44
	D-4-250	22 099.22	−4451.25	660.93	1855.18	445.30	5768.30	6334.42	5654.20	8.80	48.11	36.96
	SFT1-C25-16-2	−310.32	−1936.79	83.77	149.62	30.30	392.80	470.12	379.90	3.30	98.67	24.70
	SFT2-C25-16-2	−474.47	−2482.22	80.80	80.50	9.00	0.00	420.93	341.00	1.00	100.00	8.80
	SFT1-C50-24-3	759.64	−2493.37	130.54	1095.36	43.20	102.60	989.77	801.00	7.00	100.00	46.85
	SFT2-C50-24-3	115.81	−2850.50	104.11	336.25	20.80	61.30	801.51	645.80	4.90	100.00	32.70
	SFT1-R25-20-2	480.37	−2288.60	121.20	698.73	41.00	492.00	676.74	547.00	4.00	100.00	42.78
	SFT2-R25-20-2	−120.53	−3015.63	95.98	483.35	31.00	0.00	650.00	525.00	4.00	100.00	36.19
	SFT1-R50-30-3	303.17	−4124.09	107.40	1113.40	28.00	21.00	884.88	715.00	6.00	100.00	52.31
	SFT2-R50-30-3	−421.56	−4770.52	91.14	274.57	42.00	0.00	461.63	373.00	2.00	100.00	30.03
	SFT1-RC25-20-2	175.54	−2491.29	107.14	680.87	33.60	676.60	802.56	649.40	4.20	72.00	52.97
	SFT2-RC25-20-2	97.34	−2827.46	103.49	139.72	38.00	1.00	487.67	394.20	2.20	53.33	18.87
	SFT1-RC50-30-3	648.31	−3376.66	119.40	1809.07	15.10	704.90	1463.95	1179.70	9.70	100.00	69.51
	SFT2-RC50-30-3	1 220.40	−3177.33	140.00	1562.95	35.00	125.10	1134.30	914.60	6.80	76.61	68.67
	SFT1-R100-50-5	64.21	−6490.16	101.00	1822.23	149.80	428.50	1570.58	1268.80	10.50	80.58	50.23
	SFT2-R100-50-5	168.30	−7295.54	102.31	1581.87	61.10	274.80	1394.30	1127.70	6.60	100.00	35.73
	SFT1-R100-75-7	846.64	−9158.63	109.26	1852.04	106.30	534.60	2157.44	1740.90	15.30	87.80	47.09
	SFT2-R100-75-7	864.31	−8132.07	110.66	1891.32	47.50	1113.50	2508.02	2023.50	17.50	86.93	54.00
	Average	3 636.08	−3140.79	2 738.12	1046.00	179.45	1088.23	2270.09	1982.38	5.65	79.14	39.01

Table 15
Comparison of transportation metrics for the average values of the different SMDPDPMTWPD variants.

Problem variant	Waiting (min)	Late (min)	Distance (km)	Travel time (min)
HTW	86.65	0.00	1726.84	1492.92
HTW + CO_2	87.74	0.00	1728.42	1494.98
STW	180.72	1140.45	2304.57	2012.58
STW + CO_2	179.45	1088.23	2270.09	1982.38

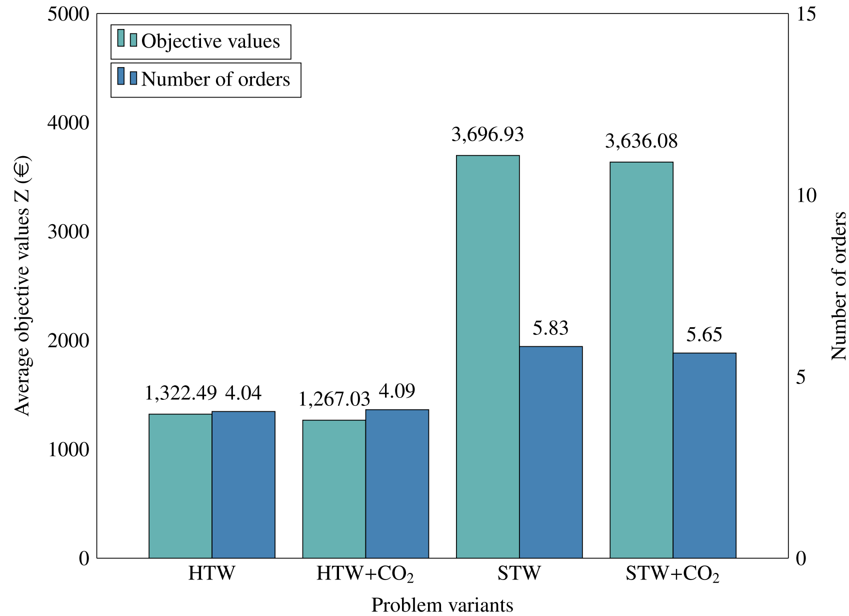

Fig. 2. Comparison of objective function values and number of orders for the different SMDPDPMTWPD variants.

A similar tendency is observed in the transportation metrics of Tables 13 and 14, when comparing the results of STW and STW + CO_2, respectively. Therefore, incorporating CO_2 emissions costs into the objective functions leads to slightly lower objective values while not heavily affecting the routing schedules. This can be explained by the relatively low CO_2 emissions costs established by the average values of the European Union's Emissions Trading Scheme, i.e., c_e = € 0.08421. We conclude that the solutions of the ALNS algorithm are beneficial for the freight-exchange-platforms profitability and align with the European Union's goal of reducing emissions.

On the other hand, when comparing the HTW to the STW and the HTW + CO_2 to the STW + CO_2, we observe a substantial change in the routing schedules. On average, with an additional 1.68 orders for the STW problem variants, the objective function values of the STWs increase more than 2.5 times the final profit compared to the HTW problem variants. This provides interesting insights. First, a slight increase in the number of orders leads to substantial additional income, demonstrating that the freight exchange platforms are a successful approach for dealing with the empty kilometers of return trips. Furthermore, the results indicate that a large profit does not require a high number of orders. Second, although companies incur penalization costs for late deliveries, the benefits of fulfilling late orders compensate for these costs, resulting in higher incomes. This is advantageous from an economic perspective. However, the impact of late deliveries should also be assessed regarding service quality. Thus, while the penalization costs for late deliveries may be low, other unmeasured costs (such as diminished customer satisfaction) could significantly affect the business case in the long term. Third, allowing trucks to fulfill late orders results in significantly higher incomes, showing the advantages of flexible delivery schedules for the profitability of the selection. Moreover, by incorporating soft time windows, companies can effectively address the issue of empty return trips and increase profits.

6. Conclusions and future work

This paper introduces the Selective Multiple Depot Pickup and Delivery Problem with Multiple Time Windows and Paired Demand (SMDPDPMTWPD). The SMDPDPMTWPD is a rich VRP variant that addresses the challenge of selecting profitable orders from a freight exchange platform to reduce backhauling costs associated with empty vehicle returns. The SMDPDPMTWPD holds significant potential for integrating transport management systems into freight exchange platforms, enabling transportation companies to reduce costs and increase revenues.

To study the SMDPDPMTWPD, we introduced four problem variants: (*i*) profit maximization, (*ii*) profit maximization incorporating CO_2 emissions costs, (*iii*) profit maximization with soft time windows, and (*iv*) profit maximization incorporating CO_2 emissions costs and soft time windows. These variants are formulated as MILPs and solved with optimization software. Given the computational complexity of the MILPs, we developed an ALNS metaheuristic to solve larger instances. We conducted six sets of experiments to test our optimization approaches and problem variants. The results indicated that ALNS outperformed a Simulated Annealing (SA) metaheuristic in 25 out of the 28 tested instances, with average objective values that are twice as large as those of SA. Furthermore, for a small instance set, the ALNS metaheuristic found the same solutions as the MILP formulation but in shorter computational times. This validates the performance of the ALNS algorithm, showing that our approximate algorithm is well-suited to solve the studied problems. Additionally, we observed that increasing the number of available orders leads to higher final profits in the optimal solutions compared to the smaller instances. This improvement is not due to vehicles carrying more orders, but because the additional orders are better suited in terms of location, time windows, and transportation costs.

Regarding the SMDPDPMTWPD variants, we observed a slight decrease in the objective function values when incorporating the CO_2 emissions costs. This tendency is observed in the results of the hard and soft time-window SMDPDPMTWPD. The results show that the SMDPDPMTWPD variants are not significantly affected by these additional costs. Moreover, we observed a significant increase in the final objective values when allowing soft time windows. Incorporating soft time windows increased the average number of orders from 4.09 to 5.65, resulting in incomes above 2.5 times larger than those for scenarios with hard time windows. These findings indicate that allowing soft time windows is crucial for improving the profitability of a selection. Despite solving instances with up to 100 orders, the vehicles served 4.90 orders on average, showing that executing a small number of strategically chosen orders is sufficient to generate substantial profits and mitigate the costs associated with empty return trips.

This work provides a foundation for several research directions. On the one hand, we believe that this tailored ALNS, along with its destroy and repair operators, can serve as a baseline for further research on algorithms for interacting with freight exchange platforms. We suggest further research on our ALNS implementation to improve the performance for large sets of instances. Similarly, valuable insights can be gained by performing a sensitivity analysis of the ALNS parameters to evaluate their impact on various problem variants and by exploring alternative methods for constructing initial solutions. On the other hand, the SMDPDPMTWPD is a problem that incorporates realistic features into the transportation system, such as multiple time windows, multiple depots, and non-mandatory order selection. In this regard, the transportation system deals with large sources of uncertainty, such as stochastic transportation times, service times, and last-minute cancellations. Consequently, future research can be conducted to develop stochastic optimization approaches to handle the uncertainty in the transportation system.

CRediT authorship contribution statement

Daniël Roelink: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Giovanni Campuzano:** Writing – review & editing, Visualization, Validation, Supervision, Software, Methodology, Investigation, Formal analysis. **Martijn Mes:** Writing – review & editing, Visualization, Supervision, Resources, Formal analysis. **Eduardo Lalla-Ruiz:** Writing – review & editing, Visualization, Validation, Supervision, Resources, Methodology, Investigation, Formal analysis, Conceptualization.

Declaration of competing interest

This is to certify that there is no conflict of interest regarding the submitted work.

Acknowledgments

The research of Dr. G. Campuzano is funded by the Chilean National Agency of Research and Development (ANID)/Scholarship Program/DOCTORADO BECAS CHILE/2019 under Grant 72200288. This financial support is gratefully acknowledged.

Appendix A. Simulated annealing metaheuristic framework

Algorithm 4 describes the Simulated Annealing (SA) metaheuristic framework implemented in the experiments of Section 5.5. The input parameters are starting (T_{start}) and ending (T_{end}) temperatures, the cooling ratio (α), the number of iterations N_{iter} at a given temperature, the counter of solutions rejected within a neighborhood structure (r), and the maximum computational time (δ_T). Additionally, $time_{init}$ and $time_{end}$ are time termination criterion handlers, x' represents a neighboring solution, T is the current annealing temperature, ϕ is a random number generated in the interval (0,1), $\mathcal{N}_l(\cdot)$ is the lth neighborhood structure, $|N|$ represents the total number of neighborhoods, and $f(\cdot)$ provides the objective value of a given solution. The calibration of the parameter values of SA is presented in Section 5.1. The neighborhood structures $\mathcal{N}_l(\cdot)$ implemented in SA are described in Appendix A.1.

The initialization phase (lines 1–4) starts by establishing the default values for the annealing temperature (line 2), the time termination handlers $time_{init}$ and $time_{end}$ (line 3), and building an initial solution that is stored in x and x^* (line 4). The heuristic procedure to build an initial solution is the same as the one implemented by ALNS (see Section 4.2). Then, the iterative phase (lines 5–19) is applied until either the maximum computational time reaches δ_T or the annealing temperature equals T (line 6). In every iteration of the main while-loop, i.e., at each given temperature T, the algorithm explores a maximum of N_{iter} solutions (lines 7–17). Thus, in every iteration of the for-loop, the SA algorithm randomly selects a neighborhood to explore (line 8), and then explores the lth neighborhood $\mathcal{N}_l(\cdot)$ until either the maximum number of rejected solutions r is reached or a feasible solution is found and stored in x' (line 9). After that, the new solution x' is stored in x if the objective function value is improved (lines 10–11). Otherwise, the SA acceptance criterion is applied (lines 12–15). Thus, a random number is generated and stored in ϕ (line 13). If $\phi < \frac{f(x')-f(x)}{T}$ the new solution is accepted and stored in x (lines 14–15). Next, x' is compared to the incumbent x^* and stored in x^* if the new objective function value outperforms the incumbent (lines 16–17). Once outside of the for-loop the annealing temperature is updated (line 18), as well as the time handler $time_{end}$ (line 19). Finally, the SA metaheuristic returns the best solution found during exploration x^*.

Algorithm 4: SA

Data: (T_{start}, T_{end}, α, r, δ_T)
1 **Initialization Phase:**
2 $T \leftarrow T_{start}$;
3 $time_{init}, time_{end} \leftarrow time()$;
4 $x, x^* \leftarrow$ **Constructive_Heuristic(**r**)**;
5 **Iterative Phase:**
6 **while** $time_{end} - time_{init} < \delta_T$ ***and*** $T > T_{end}$ **do**
7   **for** N_{iter} **do**
8         $l \leftarrow rand(|N|)$;
9         $x' \leftarrow \mathcal{N}_l(x, r)$;
10   **if** $f(x') > f(x)$ **then**
11             $x \leftarrow x'$;
12   **else**
13             $\phi \leftarrow rand(0, 1)$;
14   **if** $\phi < e^{\frac{(f(x')-f(x))}{T}}$ **then**
15                 $x \leftarrow x'$;
16   **if** $f(x') > f(x^*)$ **then**
17             $x^* \leftarrow x'$;
18     $T = T \cdot \alpha$;
19     $time_{end} \leftarrow time()$;
Result: x^*

A.1. Neighborhood structures

In this section, we list the neighborhood structures $\mathcal{N}_l(\cdot)$ implemented in the SA metaheuristic:

- **Exchange**: This neighborhood structure randomly selects an assigned order from one vehicle's route and inserts it into the route of another vehicle. The pickup location is placed first at a random position, followed by the delivery location, which is also placed randomly after the pickup to ensure route feasibility.
- **Swapping**: This neighborhood structure randomly selects two vehicles and two orders, then swaps the pickup and delivery locations of the selected orders between the two vehicle routes. In this structure, the two selected orders can belong to the same vehicle route.
- **Insertion**: This operator attempts to insert an unassigned order into one of the vehicle routes. Vehicles are explored in lexicographic order, with the pickup and delivery locations placed at random positions within the selected route.
- **Removal**: This operator randomly selects an assigned order and removes its pickup and delivery locations from the vehicle's route.

Appendix B. Comparison of metaheuristic approaches

Tables B.16, B.17, and B.18 present a comparison of ALNS and SA for the profit maximization with CO_2 emissions, the soft time windows profit maximization, and the soft time windows profit maximization

Table B.16
Average results of the ALNS and SA metaheuristic for the profit maximization with CO_2 emissions SMDPDPMTWPD.

Exp.	Data instance	ALNS				SA			
		Z (€)	Z_σ	ΔZ_{Init} (%)	Time (s)	Z (€)	Z_σ	ΔZ_{Init} (%)	Time (s)
Profit maximization with CO_2 emissions	D-1-25	**912.26**	295.15	161.09	114.13	−186.45	869.53	23.44	15.80
	D-1-50	**2763.26**	99.05	322.88	67.44	−44.05	627.41	48.66	10.08
	D-1-100	**4188.22**	60.67	552.67	478.79	2842.07	696.16	593.54	23.41
	D-1-250	**2503.01**	791.78	326.01	663.44	1500.81	1061.98	300.64	22.23
	D-2-25	282.47	222.05	159.56	79.28	**2823.97**	1907.85	344.81	72.47
	D-2-50	**2064.63**	96.59	263.57	92.16	1658.02	1837.87	407.02	16.51
	D-2-100	**3497.12**	57.16	445.19	300.11	1511.52	1302.64	154.64	97.66
	D-2-250	**5889.08**	43.80	499.88	1576.42	4139.53	2439.01	967.86	113.02
	D-4-25	27.80	131.92	101.71	128.54	**4144.22**	2155.05	2656.99	74.72
	D-4-50	**1223.24**	122.66	227.73	224.22	−40.31	1375.49	100.24	28.63
	D-4-100	2567.32	279.95	227.92	608.01	**2929.72**	3227.99	459.55	135.39
	D-4-250	**5552.79**	265.73	12 409.16	1815.13	2890.34	1577.52	196.61	82.18
	SFT1-C25-16-2	**−161.45**	7.17	76.23	108.57	−611.82	120.44	20.06	190.57
	SFT2-C25-16-2	**−474.47**	0.00	44.31	101.37	−690.26	76.65	17.56	205.65
	SFT1-C50-24-3	**733.43**	0.00	198.95	434.29	−227.31	570.84	70.72	210.60
	SFT2-C50-24-3	**126.54**	24.50	114.26	270.91	−626.77	243.87	31.36	200.80
	SFT1-R25-20-2	**372.00**	27.22	148.13	258.78	−320.81	316.34	60.76	91.38
	SFT2-R25-20-2	**−120.53**	0.00	80.00	308.71	−444.94	130.28	42.53	109.13
	SFT1-R50-30-3	**302.49**	0.00	123.64	506.83	−627.30	287.89	41.85	155.11
	SFT2-R50-30-3	**−421.56**	0.00	63.74	218.05	−765.20	127.67	36.52	161.08
	SFT1-RC25-20-2	**63.29**	8.54	108.85	276.48	−450.34	273.23	43.58	118.77
	SFT2-RC25-20-2	**93.75**	0.00	112.94	103.54	−415.10	274.94	50.94	122.66
	SFT1-RC50-30-3	**544.97**	0.00	157.97	643.69	−429.36	206.65	47.17	153.92
	SFT2-RC50-30-3	**1215.84**	0.00	553.75	624.38	−61.20	365.90	140.19	125.92
	SFT1-R100-50-5	**−6.65**	18.32	99.58	1058.85	−1072.34	321.68	35.97	233.64
	SFT2-R100-50-5	**163.12**	1.33	107.63	975.11	−1183.14	257.43	35.11	260.50
	SFT1-R100-75-7	**804.84**	15.70	138.84	1813.81	−1274.72	333.52	44.85	366.27
	SFT2-R100-75-7	**769.91**	62.23	132.75	1812.31	−1169.40	375.91	45.50	335.28
	Average	**1267.03**	93.98	641.39	559.41	492.84	834.35	250.67	133.33

Table B.17
Average results of the ALNS and SA metaheuristic for the soft time windows profit maximization SMDPDPMTWPD.

Exp.	Data instance	ALNS				SA			
		Z (€)	Z_σ	ΔZ_{Init} (%)	Time (s)	Z (€)	Z_σ	ΔZ_{Init} (%)	Time (s)
Soft time windows profit maximization	D-1-25	2 151.40	98.65	734.96	288.01	**2154.18**	83.83	463.85	56.34
	D-1-50	**4 374.31**	496.34	477.09	553.65	4149.23	586.42	473.39	68.53
	D-1-100	**5 713.38**	702.21	2172.67	1580.92	4483.24	1065.48	2138.71	139.79
	D-1-250	**11 400.9**	931.02	1108.71	1830.02	5221.18	1744.42	316.08	138.14
	D-2-25	**2 332.36**	69.69	365.11	766.82	1793.85	1672.33	293.96	933.19
	D-2-50	**4 377.11**	430.95	508.76	756.82	1072.09	2666.23	269.23	689.71
	D-2-100	**5 916.66**	330.98	496.84	1785.17	2296.80	2541.07	292.79	801.30
	D-2-250	**15 056.69**	818.46	1686.16	1850.71	2552.78	4058.61	181.49	423.34
	D-4-25	**11 697.05**	133.03	615.42	497.03	3498.05	2926.87	248.76	706.75
	D-4-50	**5 848.25**	507.06	371.90	559.77	5453.65	6615.57	407.80	851.42
	D-4-100	**8 322.08**	461.53	256.43	1383.13	6450.43	6029.66	555.88	1441.61
	D-4-250	**21 007.81**	1749.52	1514.70	1847.80	6391.77	5403.95	331.92	1311.95
	SFT1-C25-16-2	**−163.68**	11.71	92.02	93.22	−1247.89	301.03	39.67	1825.87
	SFT2-C25-16-2	**−470.83**	0.00	80.55	71.96	−1298.41	171.97	47.40	1830.10
	SFT1-C50-24-3	**823.10**	0.00	134.52	1257.33	−798.68	386.46	66.93	1846.81
	SFT2-C50-24-3	**137.72**	9.91	105.03	341.04	−1360.71	502.09	50.31	1856.73
	SFT1-R25-20-2	**496.36**	21.15	123.23	733.93	−542.98	718.43	77.30	1825.56
	SFT2-R25-20-2	**−97.67**	0.00	96.80	467.15	−1478.95	566.44	49.67	1787.09
	SFT1-R50-30-3	**348.32**	0.00	108.68	998.98	−1800.69	545.54	53.40	1836.09
	SFT2-R50-30-3	**−407.92**	0.00	91.16	281.96	−2541.71	480.81	43.62	1835.35
	SFT1-RC25-20-2	**218.73**	0.00	109.52	700.81	−974.13	464.45	59.75	1764.64
	SFT2-RC25-20-2	**115.42**	10.33	104.58	124.39	−1300.82	421.89	51.22	1817.27
	SFT1-RC50-30-3	**760.70**	13.59	123.30	1811.56	−1860.83	821.32	42.04	1835.92
	SFT2-RC50-30-3	**1 321.22**	17.11	145.89	1629.35	−815.28	1176.07	75.36	1820.59
	SFT1-R100-50-5	**124.27**	13.00	101.87	1821.86	−4130.38	849.75	38.88	1858.42
	SFT2-R100-50-5	**205.19**	40.89	102.91	1624.80	−3521.02	1125.97	50.39	1858.08
	SFT1-R100-75-7	**930.67**	13.57	109.89	1854.96	−4806.75	658.08	46.16	1880.25
	SFT2-R100-75-7	**974.40**	36.77	111.78	1856.67	−4566.10	1230.85	44.43	1887.85
	Average	**3 696.93**	247.05	430.37	1048.92	445.43	1636.27	243.23	1318.88

Table B.18
Average results of the ALNS and SA metaheuristic for the soft time windows profit maximization with CO_2 emissions SMDPDPMTWPD.

Exp.	Data instance	ALNS				SA			
		Z (€)	Z_σ	ΔZ_{Init} (%)	Time (s)	Z (€)	Z_σ	ΔZ_{Init} (%)	Time (s)
Soft time windows profit maximization with CO_2 emissions	D-1-25	**2 103.84**	73.25	216.29	290.89	2063.46	78.90	218.59	58.05
	D-1-50	**4 365.57**	523.75	630.03	579.93	4153.57	817.93	556.29	66.25
	D-1-100	**5 293.50**	672.41	839.20	1382.85	4755.95	726.71	1301.77	123.56
	D-1-250	**10 866.02**	2073.13	1 002.91	1822.73	5150.54	707.06	503.09	119.43
	D-2-25	**2 170.93**	88.38	872.63	783.08	1659.98	2645.59	190.83	762.27
	D-2-50	**4 122.57**	522.10	371.30	769.88	855.29	2527.88	1771.70	585.07
	D-2-100	**5 894.24**	151.60	1 335.15	1817.71	2560.31	3761.47	547.22	560.84
	D-2-250	**15 068.48**	868.89	1 579.17	1876.24	3041.85	3093.63	389.91	748.34
	D-4-25	**11 522.68**	270.33	506.83	498.75	6465.42	5420.24	343.47	1472.04
	D-4-50	**6 032.62**	477.71	65 031.87	565.61	3857.04	4580.54	270.49	695.71
	D-4-100	**7 853.49**	1017.51	1 912.80	1473.26	4123.17	3141.55	215.61	901.32
	D-4-250	**22 099.22**	704.64	660.93	1855.18	6373.11	4647.49	556.13	1209.93
	SFT1-C25-16-2	**−310.32**	430.29	83.77	149.62	−1239.14	460.84	42.17	1818.39
	SFT2-C25-16-2	**−474.47**	0.00	80.80	80.50	−1464.24	333.26	39.62	1835.18
	SFT1-C50-24-3	**759.64**	28.17	130.54	1095.36	−1146.37	567.72	53.89	1852.39
	SFT2-C50-24-3	**115.81**	11.88	104.11	336.25	−1646.75	415.24	42.49	1849.51
	SFT1-R25-20-2	**480.37**	0.00	121.20	698.73	−839.77	575.78	63.39	1819.47
	SFT2-R25-20-2	**−120.53**	0.00	95.98	483.35	−1518.40	457.48	49.94	1817.53
	SFT1-R50-30-3	**303.17**	0.00	107.40	1113.40	−2238.02	591.28	47.79	1859.98
	SFT2-R50-30-3	**−421.56**	0.00	91.14	274.57	−2612.44	765.39	45.26	1833.28
	SFT1-RC25-20-2	**175.54**	5.14	107.14	680.87	−1092.60	402.64	54.76	1801.88
	SFT2-RC25-20-2	**97.34**	7.57	103.49	139.72	−1215.48	398.51	54.02	1807.80
	SFT1-RC50-30-3	**648.31**	40.10	119.40	1809.07	−1595.77	730.92	53.80	1827.86
	SFT2-RC50-30-3	**1 220.40**	13.22	140.00	1562.95	−742.49	1106.22	76.77	1822.18
	SFT1-R100-50-5	**64.21**	14.68	101.00	1822.23	−3704.98	975.73	44.38	1874.93
	SFT2-R100-50-5	**168.30**	2.52	102.31	1581.87	−4587.05	954.15	37.50	1863.48
	SFT1-R100-75-7	**846.64**	36.71	109.26	1852.04	−5074.93	712.79	44.21	1877.29
	SFT2-R100-75-7	**864.31**	34.50	110.66	1891.32	−5155.22	1036.06	37.80	1877.61
	Average	**3 636.08**	288.16	2 738.12	1046.00	328.07	1522.61	273.32	1312.20

Table C.19
Computational results of the comparison between ALNS and the MILP model.

Exp.	Data instance	ALNS				ΔMILP (%)		Time (s)			
		Max	Avg	Min	σ	Avg	σ	Max	Avg	Min	σ
Profit maximization	D-1-5	−617.75	−673.33	−731.83	52.14	−9.00	8.44	4.65	4.30	4.13	0.19
	D-1-10	609.75	538.33	406.25	92.38	−11.71	15.15	26.54	25.90	25.04	0.51
	D-1-15	1155.25	1045.15	406.25	228.71	−9.53	19.80	89.60	82.18	42.09	14.18
	D-1-20	1155.25	895.35	406.25	339.69	−22.50	29.40	110.93	88.03	45.92	29.09
	D-2-5	−1313.58	−1388.67	−1427.67	51.55	−5.72	3.92	8.64	8.53	8.37	0.08
	D-2-10	−86.08	−164.43	−289.58	87.46	−91.01	101.60	35.14	33.70	31.69	1.00
	D-2-15	459.42	124.62	−289.58	358.09	−72.88	77.94	48.61	43.35	37.86	3.41
	D-2-20	459.42	256.82	−289.58	290.21	−44.10	63.17	73.31	66.06	55.41	6.68
	Average	227.71	79.23	−226.19	187.53	−33.30	39.93	49.68	44.01	31.31	6.89

with CO_2 emissions SMDPDPMTWPD, respectively. Each instance is executed 10 times, and the average results for both ALNS and SA are reported. The columns of the tables list the average objective value (Z), standard deviation (Z_σ), percentage of improvement over the initial solution (ΔZ_{Init}), and computational time in seconds. The best average objective values are bold-faced.

Appendix C. Performance evaluation of ALNS and MILP model

Table C.19 presents the results of comparing the MILP formulation and ALNS metaheuristic on small data instances. The columns report the maximum (Max), average (Avg), minimum (Min) and standard deviation (σ) objective values for the ALNS. Next, the average (Avg) and standard deviation (σ) of the percentage difference between the ALNS and MILP solutions are depicted. These are computed as $\Delta_{MILP} = \frac{Z_{MILP} - Z_{ALNS}}{Z_{MILP}} \cdot 100$, where Z_{MILP} and Z_{ALNS} represent the best solutions of the MILP formulation and ALNS metaheuristic, respectively. Finally, the maximum (Max), average (Avg), minimum (Min) and standard deviation (σ) of the run time are shown.

Appendix D. Computational details on the ALNS performance for the SMDPDPMTWPD variants

Tables D.20, D.21, D.22, and D.23 present the results of the ALNS metaheuristic for (*i*) the profit maximization (HTW), (*ii*) the profit maximization with CO_2 emissions costs (HTW + CO_2), (*iii*) the soft time windows (STW) and (*iv*) the soft time windows with CO_2 emissions costs SMDPDPMTWPD (STW + CO_2) respectively. The columns report the information for the objective values Z (€), the initial solution Z_{Init} (€), the improvement percentage (ΔZ_{Init}) of Z compared to the initial solution, and computational times (s). In these categories, we report the average (Avg) and standard deviation (σ) values and for the objective function value also the maximum (Max) and minimum (Min).

Data availability

Data will be made available on request.

Table D.20
Detailed results of the ALNS metaheuristic for the profit maximization SMDPDPMTWPD (HTW).

Exp.	Data instance	Z (€)				Z_{init} (€)		ΔZ_{Init} (%)		Time (s)	
		Max	Avg	Min	σ	Avg	σ	Avg	σ	Avg	σ
Profit maximization	D-1-25	1155.25	979.05	406.25	304.55	−203.12	770.11	154.39	80.94	112.69	33.59
	D-1-50	2908.25	2830.15	2703.75	87.65	316.01	1201.15	423.41	423.47	69.68	4.77
	D-1-100	4339.50	4268.95	4211.67	52.89	−504.78	1075.51	850.03	1046.49	507.58	43.74
	D-1-250	3145.92	2602.08	1387.33	821.66	−634.74	767.08	427.68	222.50	637.18	552.20
	D-2-25	459.42	256.82	−289.58	290.21	−1185.07	796.82	149.82	50.89	71.35	7.14
	D-2-50	2212.42	2142.69	2007.92	93.80	−1042.89	1634.26	269.38	195.57	88.82	2.31
	D-2-100	3643.67	3573.12	3515.83	52.89	−693.59	1610.16	1993.35	5291.01	280.68	8.20
	D-2-250	6053.33	5997.39	5960.50	40.58	−1957.48	1045.07	1478.04	3271.07	1811.42	7.65
	D-4-25	278.83	92.22	−128.17	151.07	−1543.20	1112.90	113.90	28.50	119.15	3.47
	D-4-50	1470.92	1302.13	1062.92	158.97	−2899.57	1570.77	147.52	20.05	210.53	3.13
	D-4-100	2902.17	2746.90	2570.83	135.70	−2733.58	879.83	216.44	58.26	583.18	10.14
	D-4-250	5934.08	5643.23	5485.17	192.66	−3389.09	969.17	284.89	76.03	1800.19	19.57
	SFT1-C25-16-2	−150.08	−156.88	−172.75	10.95	−688.70	170.88	75.53	7.99	99.86	5.63
	SFT2-C25-16-2	−470.83	−470.83	−470.83	0.00	−867.16	63.41	45.43	4.23	94.76	8.32
	SFT1-C50-24-3	773.92	773.92	773.92	0.00	−848.97	297.77	206.96	55.97	391.45	22.30
	SFT2-C50-24-3	165.92	127.62	122.25	13.51	−945.84	234.13	114.29	3.92	238.35	6.24
	SFT1-R25-20-2	451.83	396.63	382.83	29.09	−673.83	295.95	171.91	35.51	234.08	10.66
	SFT2-R25-20-2	−97.67	−97.67	−97.67	0.00	−722.59	135.17	86.00	2.94	281.90	6.26
	SFT1-R50-30-3	342.75	342.75	342.75	0.00	−879.93	293.73	144.03	18.05	464.49	21.04
	SFT2-R50-30-3	−407.92	−407.92	−407.92	0.00	−1216.03	98.60	66.26	2.76	189.05	4.84
	SFT1-RC25-20-2	120.25	106.08	100.00	9.78	−759.72	136.99	114.43	3.44	250.82	4.59
	SFT2-RC25-20-2	107.42	107.42	107.42	0.00	−654.57	219.73	118.49	7.14	93.45	2.99
	SFT1-RC50-30-3	614.00	614.00	614.00	0.00	−958.58	172.40	165.88	11.40	575.42	11.04
	SFT2-RC50-30-3	1286.50	1286.50	1286.50	0.00	−899.96	372.66	313.28	229.84	558.48	24.01
	SFT1-R100-50-5	74.50	50.98	47.75	8.37	−1757.43	219.57	102.98	0.89	935.13	13.87
	SFT2-R100-50-5	206.92	206.72	205.92	0.42	−1843.42	175.52	111.31	1.09	862.80	21.67
	SFT1-R100-75-7	941.83	910.60	904.75	11.84	−2223.03	370.52	142.27	8.88	1814.59	9.14
	SFT2-R100-75-7	939.00	804.97	596.75	92.41	−1996.33	263.53	140.80	6.15	1816.05	9.32
	Average	1407.22	1322.49	1186.79	91.39	−1228.83	605.48	308.17	398.75	542.61	31.35

Table D.21
Detailed results of the ALNS metaheuristic for the profit maximization with CO_2 emissions SMDPDPMTWPD (HTW + CO_2).

Exp.	Data instance	Z (€)				Z_{init} (€)		ΔZ_{Init} (%)		Time (s)	
		Max	Avg	Min	σ	Avg	σ	Avg	σ	Avg	σ
Profit maximization with CO_2 emissions	D-1-25	1105.96	912.26	357.46	295.15	−452.94	728.67	161.09	64.18	114.13	34.30
	D-1-50	2853.43	2763.26	2648.93	99.05	−526.86	1232.19	322.88	151.32	67.44	1.15
	D-1-100	4255.46	4188.22	4127.62	60.67	300.87	1666.20	552.67	510.46	478.79	19.79
	D-1-250	3081.55	2503.01	1340.29	791.78	−255.89	1084.21	326.01	185.23	663.44	596.58
	D-2-25	410.12	282.47	−338.37	222.05	−1194.69	758.23	159.56	68.29	79.28	6.20
	D-2-50	2157.60	2064.63	1953.10	96.59	−1715.75	1336.02	263.57	139.64	92.16	5.63
	D-2-100	3559.62	3497.12	3431.79	57.16	−1362.26	1640.95	445.19	561.62	300.11	3.91
	D-2-250	5941.85	5889.08	5849.01	43.80	−2163.96	916.31	499.88	405.76	1576.42	492.86
	D-4-25	197.52	27.80	−209.48	131.92	−2240.50	1001.75	101.71	7.43	128.54	6.49
	D-4-50	1394.72	1223.24	986.72	122.66	−2395.47	2037.72	227.73	219.25	224.22	4.71
	D-4-100	2796.74	2567.32	1865.83	279.95	−1965.62	1571.13	227.92	70.94	608.01	42.33
	D-4-250	5792.12	5552.79	5144.48	265.73	−2778.24	1942.47	12 409.16	38 161.53	1815.13	13.55
	SFT1-C25-16-2	−159.18	−161.45	−181.85	7.17	−718.15	159.64	76.23	6.81	108.57	5.10
	SFT2-C25-16-2	−474.47	−474.47	−474.47	0.00	−855.62	60.08	44.31	3.73	101.37	3.25
	SFT1-C50-24-3	733.43	733.43	733.43	0.00	−912.79	359.73	198.95	55.81	434.29	17.33
	SFT2-C50-24-3	154.62	126.54	99.39	24.50	−922.62	212.10	114.26	3.71	270.91	5.30
	SFT1-R25-20-2	423.40	372.00	354.40	27.22	−815.36	184.29	148.13	12.77	258.78	14.10
	SFT2-R25-20-2	−120.53	−120.53	−120.53	0.00	−672.92	200.83	80.00	8.09	308.71	13.60
	SFT1-R50-30-3	302.49	302.49	302.49	0.00	−1303.77	178.15	123.64	3.60	506.83	25.77
	SFT2-R50-30-3	−421.56	−421.56	−421.56	0.00	−1180.52	148.89	63.74	4.87	218.05	6.86
	SFT1-RC25-20-2	79.49	63.29	59.24	8.54	−755.18	187.42	108.85	2.32	276.48	8.40
	SFT2-RC25-20-2	93.75	93.75	93.75	0.00	−819.18	223.33	112.94	6.58	103.54	4.76
	SFT1-RC50-30-3	544.97	544.97	544.97	0.00	−1030.73	305.02	157.97	20.29	643.69	30.51
	SFT2-RC50-30-3	1215.84	1215.84	1215.84	0.00	−790.52	434.24	553.75	925.00	624.38	18.62
	SFT1-R100-50-5	40.99	−6.65	−33.00	18.32	−1672.12	231.47	99.58	1.17	1058.85	33.62
	SFT2-R100-50-5	163.95	163.12	161.20	1.33	−2162.17	224.73	107.63	0.88	975.11	10.40
	SFT1-R100-75-7	847.08	804.84	796.52	15.70	−2158.94	391.03	138.84	10.08	1813.81	7.71
	SFT2-R100-75-7	837.03	769.91	674.91	62.23	−2366.58	240.61	132.75	3.42	1812.31	8.22
	Average	1350.28	1267.03	1105.79	93.98	−1281.73	702.05	641.39	1 486.24	559.41	51.47

Table D.22
Detailed results of the ALNS metaheuristic for the soft time windows profit maximization SMDPDPMTWPD (STW).

Exp.	Data instance	Z (€)				Z_{init} (€)		ΔZ_{Init} (%)		Time (s)	
		Max	Avg	Min	σ	Avg	σ	Avg	σ	Avg	σ
Soft time windows profit maximization	D-1-25	2 234.93	2 151.40	2 019.15	98.65	−472.93	1243.92	734.96	1113.89	288.01	9.93
	D-1-50	4 783.28	4 374.31	3 672.87	496.34	−456.82	1527.30	477.09	343.02	553.65	46.59
	D-1-100	6 175.70	5 713.38	4 495.08	702.21	−111.47	1744.49	2172.67	4929.72	1580.92	391.18
	D-1-250	12 682.02	11 400.91	10 040.92	931.02	432.88	2487.89	1108.71	611.64	1830.02	24.79
	D-2-25	2 400.93	2 332.36	2 185.15	69.69	−1985.37	1196.09	365.11	361.72	766.82	60.73
	D-2-50	4 704.83	4 377.11	3 594.42	430.95	−2324.73	1659.62	508.76	803.53	756.82	62.52
	D-2-100	6 097.25	5 916.66	5 004.12	330.98	−2251.47	1720.37	496.84	281.59	1785.17	105.91
	D-2-250	16 242.73	15 056.69	14 543.18	818.46	−1937.93	1992.43	1686.16	1832.78	1850.71	30.20
	D-4-25	11 982.95	11 697.05	11 563.67	133.03	−3574.20	1773.41	615.42	491.15	497.03	106.25
	D-4-50	6 454.68	5 848.25	5 140.77	507.06	−3618.41	2736.00	371.90	328.06	559.77	46.00
	D-4-100	8 903.33	8 322.08	7 844.12	461.53	−5778.85	1854.55	256.43	44.76	1383.13	324.30
	D-4-250	23 699.88	21 007.81	19 368.65	1749.52	−3681.07	1410.89	1514.70	2791.30	1847.80	31.47
	SFT1-C25-16-2	−150.08	−163.68	−172.75	11.71	−2059.42	109.56	92.02	0.80	93.22	13.33
	SFT2-C25-16-2	−470.83	−470.83	−470.83	0.00	−2432.27	169.48	80.55	1.48	71.96	10.70
	SFT1-C50-24-3	823.10	823.10	823.10	0.00	−2411.45	271.76	134.52	3.84	1257.33	49.10
	SFT2-C50-24-3	165.92	137.72	134.58	9.91	−2775.67	257.13	105.03	0.86	341.04	48.97
	SFT1-R25-20-2	508.80	496.36	450.72	21.15	−2181.32	320.14	123.23	3.72	733.93	30.49
	SFT2-R25-20-2	−97.67	−97.67	−97.67	0.00	−3058.59	172.83	96.80	0.18	467.15	21.70
	SFT1-R50-30-3	348.32	348.32	348.32	0.00	−4039.08	344.14	108.68	0.68	998.98	43.16
	SFT2-R50-30-3	−407.92	−407.92	−407.92	0.00	−4641.04	378.93	91.16	0.76	281.96	59.28
	SFT1-RC25-20-2	218.73	218.73	218.73	0.00	−2331.43	296.99	109.52	1.23	700.81	50.80
	SFT2-RC25-20-2	127.42	115.42	107.42	10.33	−2560.61	343.57	104.58	0.74	124.39	23.07
	SFT1-RC50-30-3	767.12	760.70	726.23	13.59	−3301.58	353.74	123.30	2.74	1811.56	6.47
	SFT2-RC50-30-3	1 326.63	1 321.22	1 272.52	17.11	−2917.40	373.51	145.89	5.27	1629.35	126.73
	SFT1-R100-50-5	145.27	124.27	104.53	13.00	−6647.26	384.94	101.87	0.21	1821.86	14.14
	SFT2-R100-50-5	221.72	205.19	89.15	40.89	−7117.25	437.00	102.91	0.63	1624.80	185.71
	SFT1-R100-75-7	947.73	930.67	913.02	13.57	−9428.40	394.54	109.89	0.40	1854.96	28.58
	SFT2-R100-75-7	1 055.37	974.40	929.73	36.77	−8296.17	494.53	111.78	0.79	1856.67	35.69
	Average	3 996.15	3 696.93	3 372.89	247.05	−3284.26	944.64	430.37	498.48	1048.92	70.99

Table D.23
Detailed results of the ALNS metaheuristic for the soft time windows profit maximization with CO_2 emissions SMDPDPMTWPD (STW + CO_2).

Exp.	Data instance	Z (€)				Z_{init} (€)		ΔZ_{Init} (%)		Time (s)	
		Max	Avg	Min	σ	Avg	σ	Avg	σ	Avg	σ
Soft time windows profit maximization with CO_2 emissions	D-1-25	2 153.62	2 103.84	1 937.84	73.25	44.18	1308.70	216.29	123.57	290.89	4.75
	D-1-50	4 713.47	4 365.57	3 608.04	523.75	−713.81	1410.04	630.03	582.60	579.93	11.62
	D-1-100	6 069.25	5 293.50	4 404.20	672.41	−69.07	1534.14	839.20	914.68	1382.85	372.14
	D-1-250	12 589.78	10 866.02	5 411.38	2073.13	−190.35	1585.85	1 002.91	802.93	1822.73	12.46
	D-2-25	2 273.27	2 170.93	2 057.49	88.38	−732.08	1228.27	872.63	1 812.28	783.08	63.95
	D-2-50	4 628.21	4 122.57	3 522.79	522.10	−1364.37	1547.94	371.30	217.33	769.88	33.19
	D-2-100	5 984.00	5 894.24	5 552.92	151.60	−1258.48	2056.14	1 335.15	2 895.80	1817.71	8.95
	D-2-250	16 078.04	15 068.48	14 395.43	868.89	−1934.77	1245.82	1 579.17	1 727.99	1876.24	34.61
	D-4-25	11 809.23	11 522.68	11 072.53	270.33	−3087.20	907.39	506.83	136.60	498.75	112.98
	D-4-50	6 323.25	6 032.62	5 014.32	477.71	−3775.13	2168.38	65 031.87	204 749.56	565.61	37.02
	D-4-100	8 834.78	7 853.49	6 205.03	1017.51	−3498.85	2084.26	1912.80	4 945.19	1473.26	296.61
	D-4-250	23 436.02	22 099.22	21 753.42	704.64	−4451.25	1487.98	660.93	226.04	1855.18	26.44
	SFT1-C25-16-2	−159.18	−310.32	−1 534.58	430.29	−1936.79	161.86	83.77	22.68	149.62	193.51
	SFT2-C25-16-2	−474.47	−474.47	−474.47	0.00	−2482.22	175.69	80.80	1.35	80.50	10.84
	SFT1-C50-24-3	773.00	759.64	706.20	28.17	−2493.37	151.60	130.54	1.71	1095.36	48.66
	SFT2-C50-24-3	133.23	115.81	108.43	11.88	−2850.50	255.86	104.11	0.73	336.25	42.41
	SFT1-R25-20-2	480.37	480.37	480.37	0.00	−2288.60	247.00	121.20	2.20	698.73	35.07
	SFT2-R25-20-2	−120.53	−120.53	−120.53	0.00	−3015.63	222.52	95.98	0.29	483.35	23.55
	SFT1-R50-30-3	303.17	303.17	303.17	0.00	−4124.09	363.51	107.40	0.65	1113.40	53.91
	SFT2-R50-30-3	−421.56	−421.56	−421.56	0.00	−4770.52	278.57	91.14	0.52	274.57	38.16
	SFT1-RC25-20-2	177.97	175.54	165.79	5.14	−2491.29	278.31	107.14	0.95	680.87	38.21
	SFT2-RC25-20-2	111.70	97.34	93.75	7.57	−2827.46	342.21	103.49	0.53	139.72	24.12
	SFT1-RC50-30-3	666.47	648.31	548.40	40.10	−3376.66	333.68	119.40	2.45	1809.07	6.32
	SFT2-RC50-30-3	1 230.49	1 220.40	1 201.27	13.22	−3177.33	624.17	140.00	9.22	1562.95	63.02
	SFT1-R100-50-5	95.48	64.21	49.54	14.68	−6490.16	292.98	101.00	0.27	1822.23	14.50
	SFT2-R100-50-5	170.63	168.30	164.50	2.52	−7295.54	347.42	102.31	0.12	1581.87	168.97
	SFT1-R100-75-7	900.46	846.64	787.16	36.71	−9158.63	474.17	109.26	0.56	1852.04	36.27
	SFT2-R100-75-7	935.59	864.31	831.73	34.50	−8132.07	499.79	110.66	0.74	1891.32	27.32
	Average	3 917.70	3 636.08	3 136.59	288.16	−3140.79	843.37	2 738.12	7 827.84	1046.00	65.70

References

[1] CBS. 2023, https://opendata.cbs.nl/#/CBS/nl/dataset/83101NED/table?ts=1712140961134. [Accessed 03 April 2024].

[2] CBS. 2024, https://opendata.cbs.nl/#/CBS/nl/dataset/83760NED/table?ts=1712142139308. [Accessed 03 April 2024].

[3] CBS. 2024, https://opendata.cbs.nl/#/CBS/nl/dataset/83077NED/table?ts=1712140231380. [Accessed 03 April 2024].

[4] Sar K, Ghadimi P. A systematic literature review of the vehicle routing problem in reverse logistics operations. Comput Ind Eng 2023;177:109011.

[5] El Bouyahyiouy K, Bellabdaoui A. The selective full truckload multi-depot vehicle routing problem with time windows: Formulation and a genetic algorithm. Int J Supply Oper Manag 2022;9(3):299–320.

[6] Li Y, Yang J. The last-mile delivery vehicle routing problem with handling cost in the front warehouse mode. Comput Ind Eng 2024;190:110076.
[7] El Bouyahyiouy K, Bellabdaoui A. A new crossover to solve the full truckload vehicle routing problem using genetic algorithm. In: 2016 3rd international conference on logistics operations management. GOL, IEEE; 2016, p. 1–6.
[8] El Bouyahyiouy K, Bellabdaoui A. An ant colony optimization algorithm for solving the full truckload vehicle routing problem with profit. In: 2017 international colloquium on logistics and supply chain management. LOGISTIQUA, IEEE; 2017, p. 142–7.
[9] El Bouyahyiouy K, Bellabdaoui A. A mixed-integer linear programming model for the selective full-truckload multi-depot vehicle routing problem with time windows. Decis Sci Lett 2021;10(4):471–86.
[10] El Bouyahyiouy K, Bellabdaoui A. An am-TSPTW transformation and a RTS algorithm for commodity selection and vehicle routing planning in full truckload industry. In: 2022 14th international colloquium of logistics and supply chain management. LOGISTIQUA, IEEE; 2022, p. 1–6.
[11] El Bouyahyiouy K, Bellabdaoui A. A genetic-based algorithm for commodity selection and full truckload vehicle routing problem. In: Digital technologies and applications: proceedings of ICDTA'23, Fez, Morocco, volume 2. Springer; 2023, p. 806–16.
[12] Archetti C, Speranza MG, Vigo D. Chapter 6: Pickup-and-delivery problems for goods transportation. In: Vehicle routing: problems, methods, and applications, second edition. SIAM; 2014, p. 161–92.
[13] Verdonck L. Collaborative logistics from the perspective of freight transport companies (Ph.D. thesis), Universiteit Hasselt; 2017, Available at https://documentserver.uhasselt.be/bitstream/1942/23685/1/DoctoraatLotteVerdonck.pdf.
[14] Dumas Y, Desrosiers J, Soumis F. The pickup and delivery problem with time windows. European J Oper Res 1991;54(1):7–22.
[15] Jung S, Haghani A. Genetic algorithm for a pickup and delivery problem with time windows. Transp Res Rec 2000;1733(1):1–7.
[16] Alaïa EB, Dridi IH, Bouchriha H, Borne P. Optimization of the multi-depot & multi-vehicle pickup and delivery problem with time windows using genetic algorithm. In: 2013 international conference on control, decision and information technologies. coDIT, IEEE; 2013, p. 343–8.
[17] Ropke S, Pisinger D. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transp Sci 2006;40(4):455–72.
[18] Krajewska MA, Kopfer H, Laporte G, Ropke S, Zaccour G. Horizontal cooperation among freight carriers: request allocation and profit sharing. J Oper Res Soc 2008;59(11):1483–91.
[19] Dahl S, Derigs U. Cooperative planning in express carrier networks—An empirical study on the effectiveness of a real-time decision support system. Decis Support Syst 2011;51(3):620–6.
[20] Baldacci R, Bartolini E, Mingozzi A. An exact algorithm for the pickup and delivery problem with time windows. Oper Res 2011;59(2):414–26.
[21] Heilig L, Lalla-Ruiz E, Voß S. Port-IO: an integrative mobile cloud platform for real-time inter-terminal truck routing optimization. Flex Serv Manuf J 2017;29(3–4):504–34.
[22] Adi TN, Bae H, Iskandar YA. Interterminal truck routing optimization using cooperative multiagent deep reinforcement learning. Processes 2021;9(10):1728.
[23] Alaïa EB, Dridi IH, Bouchriha H, Borne P. A particle swarm optimization for the multi-depots pick-up and delivery problems with time windows and multi-vehicles. In: International conference on industrial engineering and systems management IESM. 2017, p. 623–8.
[24] Harbaoui Dridi I, Ben Alaïa E, Borne P, Bouchriha H. Optimisation of the multi-depots pick-up and delivery problems with time windows and multi-vehicles using PSO algorithm. Int J Prod Res 2020;58(14):4201–14.
[25] Ting C-K, Liao X-L. The selective pickup and delivery problem: Formulation and a memetic algorithm. Int J Prod Econ 2013;141(1):199–211.
[26] Ting C-K, Liao X-L, Huang Y-H, Liaw R-T. Multi-vehicle selective pickup and delivery using metaheuristic algorithms. Inform Sci 2017;406:146–69.
[27] Al Chami Z, Manier H, Manier M-A. New model for a variant of pick up and delivery problem. In: 2016 IEEE international conference on systems, man, and cybernetics. SMC, IEEE; 2016, p. 001708–13.
[28] Al Chami Z, Manier H, Manier M-A. A lexicographic approach for the bi-objective selective pickup and delivery problem with time windows and paired demands. Ann Oper Res 2017;273:237–55.
[29] Al Chami Z, Manier H, Manier M-A, Fitouri C. A hybrid genetic algorithm to solve a multi-objective pickup and delivery problem. IFAC- Pap 2017;50(1):14656–61.
[30] Al Chami Z, El Flity H, Manier H, Manier M-A. A new metaheuristic to solve a selective pickup and delivery problem. In: 2018 4th international conference on logistics operations management. GOL, IEEE; 2018, p. 1–5.
[31] Al Chami Z, Manier H, Manier M-A. A new compact two-index formulation for a pickup and delivery problem. 2018, Available at https://www.openscience.fr/IMG/pdf/iste_ro18v1n1_1.pdf.
[32] Al Chami Z, El Flity H, Manier H, Manier M-A. Multi-period pickup and delivery problem with time windows and paired demands. In: 2018 15th international conference on control, automation, robotics and vision. ICARCV, IEEE; 2018, p. 337–42.
[33] Al Chami Z, Bechara B, Manier H, Manier M-A. A robust pickup and delivery problem with uncertain travel time. In: 2018 IEEE 30th international conference on tools with artificial intelligence. ICTAI, IEEE; 2018, p. 940–6.
[34] Al Chami Z, Bechara B, Manier H, Manier M-A, Sleiman M. A GRASP-ALNS combination for robust pickup and delivery problem. Int J Prod Res 2022;60(12):3809–28.
[35] Li Y, Chen H, Prins C. Adaptive large neighborhood search for the pickup and delivery problem with time windows, profits, and reserved requests. European J Oper Res 2016;252(1):27–38.
[36] Gansterer M, Küçüktepe M, Hartl RF. The multi-vehicle profitable pickup and delivery problem. Or Spectr 2017;39:303–19.
[37] Peng Z, Al Chami Z, Manier H, Manier M-A. A particle swarm optimization for selective pickup and delivery problem. In: 2018 IEEE 30th international conference on tools with artificial intelligence. ICTAI, IEEE; 2018, p. 947–52.
[38] Peng Z, Al Chami Z, Manier H, Manier M-A. A hybrid particle swarm optimization for the selective pickup and delivery problem with transfers. Eng Appl Artif Intell 2019;85:99–111.
[39] Sun P, Veelenturf LP, Hewitt M, Van Woensel T. Adaptive large neighborhood search for the time-dependent profitable pickup and delivery problem with time windows. Transp Res Part E: Logist Transp Rev 2020;138:101942.
[40] Ben-Said A, Moukrim A, Guibadj RN, Verny J. Using decomposition-based multi-objective algorithm to solve selective pickup and delivery problems with time windows. Comput Oper Res 2022;145:105867.
[41] Baklagis D, Dikas G, Minis I. The team orienteering pick-up and delivery problem with time windows and its applications in fleet sizing. RAIRO- Oper Res- Rech Opér 2016;50(3):503–17.
[42] Liu C, Aleman DM, Beck JC. Modelling and solving the senior transportation problem. In: Integration of constraint programming, artificial intelligence, and operations research: 15th international conference, CPAIOR 2018, Delft, the Netherlands, June 26–29, 2018, proceedings 15. Springer; 2018, p. 412–28.
[43] Santini A, Plum CE, Ropke S. A branch-and-price approach to the feeder network design problem. European J Oper Res 2018;264(2):607–22.
[44] Nadizadeh A. Formulation and a heuristic approach for the orienteering location-routing problem. RAIRO- Oper Res 2021;55:S2055–69.
[45] Eurosender. How to calculate LDM for shipping? 2023, https://www.eurosender.com/blog/en/calculate-ldm-shipping/. [Accessed 31 May 2024].
[46] HST Groep. Logistiek Jargon. 2024, https://www.hst.nl/logistiek-jargon/. [Accessed 31 May 2024].
[47] Schulte F, Lalla-Ruiz E, González-Ramírez RG, Voß S. Reducing port-related empty truck emissions: a mathematical approach for truck appointments with collaboration. Transp Res Part E: Logist Transp Rev 2017;105:195–212, URL https://www.sciencedirect.com/science/article/pii/S1366554516300801.
[48] Fernández Gil A, Lalla-Ruiz E, Gómez Sánchez M, Castro C, et al. A review of heuristics and hybrid methods for green vehicle routing problems considering emissions. J Adv Transp 2022;2022.
[49] Corona-Gutiérrez K, Nucamendi-Guillén S, Lalla-Ruiz E. Vehicle routing with cumulative objectives: a state of the art and analysis. Comput Ind Eng 2022;169:108054.
[50] Azuero-Ortiz J, Gaviria-Hernández M, Jiménez-Rodríguez V, Vale-Santiago E, González-Neira E. Design of a hybridization between tabu search and PAES algorithms to solve a multi-depot, multi-product green vehicle routing problem. Decis Sci Lett 2023;12(2):441–56.
[51] Mara STW, Norcahyo R, Jodiawan P, Lusiantoro L, Rifai AP. A survey of adaptive large neighborhood search algorithms and applications. Comput Oper Res 2022;105903.
[52] Azi N, Gendreau M, Potvin J-Y. An adaptive large neighborhood search for a vehicle routing problem with multiple routes. Comput Oper Res 2014;41:167–73.
[53] Avci MG, Avci M. An adaptive large neighborhood search approach for multiple traveling repairman problem with profits. Comput Oper Res 2019;111:367–85.
[54] Oda T, Liu Y, Sakamoto S, Elmazi D, Barolli L, Xhafa F. Analysis of mesh router placement in wireless mesh networks using friedman test considering different meta-heuristics. Int J Commun Netw Distrib Syst 2015;15(1):84–106.
[55] Pereira DG, Afonso A, Medeiros FM. Overview of Friedman's test and post-hoc analysis. Comm Statist Simulation Comput 2015;44(10):2636–53.
[56] Boyacı B, Dang TH, Letchford AN. Vehicle routing on road networks: how good is euclidean approximation? Comput Oper Res 2021;129:105197.
[57] European Commission. 2023, https://road-safety.transport.ec.europa.eu/eu-road-safety-policy/priorities/safe-road-use/safe-speed/archive/current-speed-limit-policies_en. [Accessed 13 July 2023].
[58] Rietveld P, Zwart B, Van Wee B, van den Hoorn T. On the relationship between travel time and travel distance of commuters. Ann Reg Sci 1999;33:269–87.
[59] Hornstra RP, Silva A, Roodbergen KJ, Coelho LC. The vehicle routing problem with simultaneous pickup and delivery and handling costs. Comput Oper Res 2020;115:104858.
[60] Taş D, Dellaert N, Van Woensel T, De Kok T. Vehicle routing problem with stochastic travel times including soft time windows and service costs. Comput Oper Res 2013;40(1):214–24.