The ground handler dock capacitated pickup and delivery problem with time windows
A collaborative framework for air cargo operations

Bombelli, A.; Fazi, Stefano

DOI
10.1016/j.tre.2022.102603

Publication date
2022

Document Version
Final published version

Published in
Transportation Research. Part E: Logistics and Transportation Review

Transportation Research Part E 159 (2022) 102603

Contents lists available at ScienceDirect

Transportation Research Part E

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

The ground handler dock capacitated pickup and delivery problem with time windows: A collaborative framework for air cargo operations

Alessandro Bombelli [a,*], Stefano Fazi [b]

[a] Delft University of Technology, Air Transport and Operations, Faculty of Aerospace Engineering, P.O. Box 5015, 2600 GA Delft, The Netherlands

[b] Delft University of Technology, Department of Technology, Policy and Management, P.O. Box 5015, 2600 GA Delft, The Netherlands

ARTICLE INFO

Keywords:
Air cargo supply chain
Pickup and delivery problem
Scheduling
Waiting times
Centralized planning
Collaboration

ABSTRACT

We study a typical problem within the air cargo supply chain, concerning the transportation of standard Unit Load Devices (ULDs) from freight forwarders' to ground handlers' warehouses. First, ULDs are picked up by a set of available trucks at the freight forwarders' premises within a time window. Next, they are delivered to the ground handlers, also within a time window, and discharged according to a Last In First Out (LIFO) policy. Due to space constraints, ground handlers have limited capacity to serve the trucks and waiting times may arise, especially in case freight forwarders do not coordinate their operations. Therefore, in this paper we consider a cooperative framework where this transportation is coordinated by a central planner. The goal of the planner is to find a proper routing and scheduling that minimizes the sum of the transportation and waiting times at the ground handlers' warehouses, while satisfying the capacity of the trucks. We propose two mathematical formulations, one based on the routing and the other based on the packing aspect of the problem. To solve large instances of the problem, an Adaptive Large Neighborhood Search algorithm is also developed. With numerical experiments, we compare the performances of the two models and the metaheuristic, and we quantify the benefits of the proposed framework to reduce waiting times.

1. Introduction

Air cargo business represents a large share of airlines' income, comparable to the one from first class passengers (Drljača, 2017). As a consequence of the recent reduction of passengers' volumes due to the COVID-19 pandemic, this business is more and more vital for the industry. Even though the volumes are low compared to other forms of cargo transportation, air cargo has witnessed a steady increase in the last two decades and a focus on both valuable and perishable products, for which this transportation mode is essential to ease international trades. In addition, it is the only transportation mode that guarantees a seamless worldwide flow of cargo when time is a crucial factor (IATA, 2020).

The air cargo supply chain is a complex multi-modal and multi-stakeholder supply chain. In general, shippers delegate the delivery of their goods to a freight forwarder (FF), that is a specialized company that organizes the transportation to the final point of distribution (i.e., consignee). Especially larger FFs might already consolidate a considerable percentage of their shipments in-house into Unit Load Devices (ULDs), which are standardized containers for air cargo (Ankersmit et al., 2014). Next, these ULDs

* Corresponding author.
E-mail addresses: a.bombelli@tudelft.nl (A. Bombelli), s.fazi@tudelft.nl (S. Fazi).

https://doi.org/10.1016/j.tre.2022.102603
Received 4 June 2021; Received in revised form 20 November 2021; Accepted 1 January 2022
Available online 7 February 2022
1366-5545/

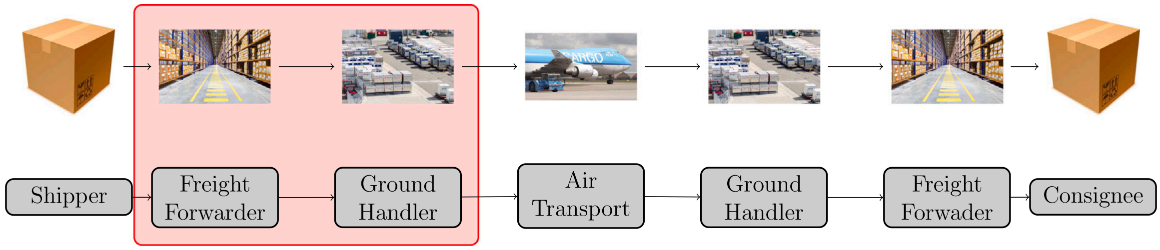

Fig. 1. Representation of a typical air cargo supply chain, with the scope of this paper highlighted. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

are transported via truck to a second set of warehouses, located at the airport, where ground handlers (GHs) take care of distributing and loading the cargo onto the aircraft. After the air transport leg, the reverse process occurs at the destination airport, until goods are delivered to the intended consignee. In addition to the stakeholders already introduced, FFs might rely on third party logistics service providers for ground transport to/from GHs. A graphical representation of the overall supply chain is depicted in Fig. 1, with the focus of this work highlighted in red.

In cargo airports, there is generally a strong supply/demand imbalance between the volume of trucks arriving to a GH warehouse and its actual processing capabilities. This imbalance is due to the fact that, even in major cargo hub airports, the number of GHs is limited. The consequences are congestion and waiting times, especially during the cargo peak season and peak days (e.g., Fridays), also due to uncoordinated arrivals of trucks from FFs (Vracken, 2020). The risk is to create bottlenecks and exceed ULDs due dates, which may generate large rebooking costs (Liu et al., 2019; Niu et al., 2019). For this reason, similarly to other industries (Gansterer and Hartl, 2018), more and more airports are favoring coordination between FFs and GHs to improve their operations. For example, Bruxelles Airport has recently introduced an online slot booking system, where FFs can book slots to pick up/deliver shipments at/to the intended GHs (Nallian, 2021). Jan de Rijk, one of Europe's largest trucking company, is also advocating for greater cooperation between airport stakeholders in a bid to reduce congestion (The loadstar, 2021). At Amsterdam Airport Schiphol, collaboration initiatives such as the "Milkrun" (Milkrun, 2021) or the "Drop & Collect" (Schiphol, 2021) are being currently tested conceptually or via pilots. In the literature, some of the aforementioned initiatives have been assessed with different methods such as discrete event simulation (Ankersmit et al., 2014; Buso, 2017), system dynamics (Cheung and Wong, 2015) and multi criteria decision analysis (Rezaei et al., 2017). However, the literature still lacks studies that address cooperation schemes in the landside air cargo supply chain from an operational perspective. Our work aims to fill this gap.

In this paper, we evaluate a cooperative logistics system within this part of the air cargo supply chain. In particular, we introduce cooperation in the form of a consortium of FFs that uses a shared neutral fleet of trucks (gray fleet) replacing trucks controlled by each FF independently. A central planner, with full access to information regarding shipments to be delivered to GHs, is in charge of the routing, loading, and dock assignment strategy of the gray fleet. The problem can be modeled as a new variant of the vehicle routing problem with pickup and deliveries, time windows and loading constraints. In particular, a set of capacitated trucks departs from a central depot and visits the FFs to pickup shipments, packed in ULDs. Next, GHs' warehouses, with a limited number of docking stations, are visited to unload the shipments following a Last In First Out (LIFO) strategy, since the ULDs' width is typically as large as the trucks' rear and can only be moved longitudinally; see Fig. 2 for an illustration. The goal is to minimize routing times and trucks' waiting times due to scheduling at the docks. We name the problem as Ground Handler Dock-Capacitated Pickup and Delivery Problem with Time Windows (GHDC-PDPTW). The novelty of the model is about combining the routing decision with the scheduling of the trucks and the ULDs discharge at the GHs' docks to reduce waiting times, and the enforcement of the LIFO strategy which affects both routing and scheduling.

The aim of this study is threefold. First, we model the proposed problem by developing two mathematical Mixed Integer Linear Programming (MILP) formulations; the first with a full routing problem approach and the other with a hybrid bin packing and routing problem approach. Second, we propose an Adaptive Large Neighborhood Search (ALNS) heuristic to solve large instances of the problem. Third, by means of an experimental framework, we compare the performances of the MILP formulations against the metaheuristic, generate insights on the benefits of the cooperative scheme, and finally provide a sensitivity analysis on the main parameters of the problem.

This paper is structured as follows. In Section 2, we will position our work in the available literature on air cargo logistics and relevant classical models. In Section 3, we will formally describe the problem and develop the mathematical formulations. Section 4 will show the metaheuristic approach. In Section 5, we will present the experimental framework and the results along with a discussion. Section 6 will conclude the paper with our final remarks and possible future directions.

2. Literature review

2.1. Air cargo operations

Literature on air cargo operations has grown substantially in the last three decades, in response to the new challenges for the industry to remain competitive in the market. Moreover, air cargo transport entails more complex problems than passenger

Fig. 2. Unloading of a ULD from a trailer (left) and focus on the trailer's base with blocks and guides (right). From https://www.stattimes.com and https://www.jbtc.com respectively. Potentially, trailers may be accessible from the side thanks to removable covers (Fig. 2(a)). However, blocks and guides (Fig. 2(b)) impose that the forklifts can only unload ULDs following a LIFO sequence.

transport due to more elaborated processes, different levels of integration and consolidation, and multiple options in several decisions (Bartodziej et al., 2009). For an overview of the literature up to 2015, we refer to the excellent work of Feng et al. (2015). The review classifies the papers based on the three main perspectives in the air cargo industry: airlines, service supply chains and freight forwarders. In this section, we will closely follow this classification and will review also more recent work related to our study.

With regard to airlines, literature is vast and covers four main areas: revenue management; terminal operations; fleet routing and flight scheduling; aircraft loading. In "revenue management", most works tackle the problem of overbooking; among others, see the papers of Amaruchkul et al. (2007) for accept/reject decisions of freight forwarders' requests and Qin et al. (2012) for a dynamic programming model determining inventory control considering overbooking. "Terminal operations" consist of all planning and scheduling for manpower capacity and trucks within the airport area. With regard to the latter, Ou et al. (2010) analyze truck arrivals and unloading aiming to minimize handling and storage costs; whereas (Xu et al., 2014) tackle cargo routing and scheduling aiming to minimize processing and congestion costs. Finally, for "fleet routing and flight scheduling" and "aircraft loading" the focus is on the aircraft management, including crew scheduling (Schaefer et al., 2005), ULDs loading onto the aircraft (Roesener and Barnes, 2016; Lurkin and Schyns, 2015), route construction (Gebhardt et al., 2015).

The air cargo service supply chains perspective concerns strategic decisions, such as coordination, and competition. These items have been investigated only in a few works. See for example, system radical improvements by means of business process reengineering techniques (Khan, 2000); new digital forms of integration between parties (Leung et al., 2000).

Freight forwarders are the core element of this research. Before 2015 their role did not receive much attention (Feng et al., 2015). Relevant problems are: capacity booking, container loading, integration and consolidation strategies, and truck routing and scheduling. In terms of "container loading", literature has focused on the packing of ULDs, whereas in our work this decision is an input of the model encapsulated in the information on weight and time windows. See Wu (2010) for decisions on rental of containers based on volume and weight, and Huang and Chi (2007) for consolidation decisions to minimize the fees charged by the airlines. With respect to truck scheduling and pickup and delivery decisions, the real-world problems are not entirely unique and they are already included in the available variants of routing problem. Literature is quite scarce in this setting, yet the following contributions are relevant. Leung et al. (2017) take a tactical perspective, tackling the problem of allotment booking. First, the FF needs to book a set of resources (ULDs, transport services, etc.). Secondly, these resources are allocated to the individual shipments. The authors solve the first part with a two-stage stochastic dynamic program and the second with a heuristic approach. More recently, Huang et al. (2020) study the inbound supply chain from the shipper to the final destination. Their problem includes consolidation decisions into ULD in order to minimize trucking costs once the cargo has landed. The problem is modeled with a bin packing type of model and solved with a solution algorithm based on Lagrangian Relaxation. In terms of cooperation schemes between FFs, Lai et al. (2019) study the possibility of capacity sharing for spot market requests with an efficient auction strategy. Finally, in Archetti and Peirano (2020) and Angelelli et al. (2020), an exact formulation and a math-heuristic are respectively presented to address the Air Transportation Freight Forwarder Service Problem (ATFFSP). In the ATFFSP, the goal is to minimize the overall transportation costs from origin to destination for a set of shipments, using the perspective of a FF. Given the complexity of the problem, possible congestion issues on the ground due to limited capacity are not explicitly considered.

Considering works addressing more localized operational improvements of the air cargo supply chain, it is clear from this review that most models address airside problems, whereas there is a lack of operational models addressing the transport of goods from FFs to GHs (and vice versa).

With regard to cooperation schemes in air cargo supply chains, literature is quite scarce. One of the few attempts can be found in Van Alebeek and Bombelli (2021), where collaboration and competition between FFs are simultaneously accounted for by means of *coopetition* (portmanteau for the two words). The work is inspired by Berger and Bierwirth (2010), where the authors propose two solution approaches for requests' reassignment, involving decentralized control and auction based exchange mechanisms. For an extensive literature review on collaboration schemes in vehicle routing problems, we refer to Gansterer and Hartl (2018). In other fields, such as urban distribution, consolidation and collaboration schemes have been analyzed with different perspectives. With regard to consolidation, Simoni et al. (2018) analyze different city logistics solutions for consolidation of parcels. They propose an extended multi-depot vehicle routing problem where the main decisions are whether to use a facility or not and the number of certain vehicle types. However, the paper considers only one single large delivery company and suggests to investigate the dynamics when multiple stakeholders collaborate. Regarding collaboration schemes, Hezarkhani et al. (2019) and Ciardiello et al. (2021) develop game theoretical models to calculate, respectively, gain and cost allocations.

2.2. Related pickup and delivery models

From a modeling perspective, our work is a variant of the one-to-one pickup and delivery problem (PDP) where each demand has a specified origin and destination (Berbeglia et al., 2007). Readers are referred to Parragh et al. (2008), Berbeglia et al. (2007) and Lin et al. (2014), Adewumi and Adeleke (2018), Braekers et al. (2016) for exhaustive surveys on PDP and VRP models, respectively. In addition, readers are referred to Iori and Martello (2010) for a review of models where vehicle routing and loading constraints (two- or even three-dimensional) are simultaneously accounted for.

A key component of our problem is the presence of LIFO constraints, which is not new in the literature, yet only recently studied. Early work from Levitin and Abezgaouz (2003) addresses the problem of finding the shortest paths of multiple AGVs that pickup and deliver pallets according to a LIFO rule. Cordeau et al. (2010) considers a traveling salesman problem where each pickup location has an associated destination and with LIFO constraints. A branch-and-cut algorithm is proposed. Cherkesly et al. (2015) study the routing problem with origin and destination depots, where each pickup and delivery location has an associated time window and solve the problem using branch-and-price-and-cut algorithms. Benavent et al. (2015) extend the model by including a maximum time duration for the route and propose two formulations and a heuristic algorithm based on Tabu Search. Cherkesly et al. (2016) further extend these works by considering stacks at the rear of the trucks.

Compared to these contributions, our problem considers that pickup operations must all be performed before the delivery ones. However, the main difference is also the presence of an extra decision layer concerning scheduling decisions analogous to parallel machine scheduling problems. This relates to the allocation of trucks to the docks of each GH's warehouse and the presence of scheduling constraints that avoid conflicts between trucks. In fact, the delivery nodes are typically not capacitated and the discharge can occur at any time independently of the presence of other trucks at the same time at the dock. In our case, this is a critical aspect due to limited capacity at the GHs' side and that relates to some extent to the literature on cross-docking operations (see Ladier and Alpan, 2016 for a recent review). In Lee et al. (2006) a VRP is modeled to impose that pickup and delivery trucks reach the cross-dock facility simultaneously to avoid waiting times. In Konur and Golias (2013) arrival times of trucks at a cross-dock with limited number of docks are not known and the problem is to schedule the trucks properly based on available information to reduce waiting times. Rijal et al. (2019) propose the so-called cross dock scheduling problem where two decisions are considered: the location decision called truck-to-door assignment problem to minimize traveled distance and the timing decision called the truck scheduling problem to minimize waiting time. A performing Adaptive Large Neighborhood Search algorithm is developed for resolution. Finally, Dondo and Cerdá (2015) is, to our knowledge, the only attempt in this field of research that integrates routing, dock assignment and scheduling decisions. Contrarily to our problem, the paper considers the cross dock as start/end node and the only node with scarcity of docks. Moreover, although pickup requests must be accomplished before delivery ones, LIFO constraints are not applied. A mixed-integer linear programming formulation is proposed and solved using a commercial solver. All in all, we can conclude that the proposed problem provides a unique setting that has not received attention in the present literature.

3. Problem formulation

In this section, we first define the problem. Next, we introduce the general mathematical notation and, finally, we present the two MILP formulations. The first is based on the routing aspect of the problem and is used later in the paper as a basis for the metaheuristic. The second is a compact formulation that proved to be more performing in a preliminary experimental phase.

3.1. Problem setting

We focus on the export transportation of cargo from a consortium of FFs to GHs. A central planner, managing the gray fleet, receives a list of shipments from the consortium concerning the planning horizon of interest. The shipments come in the form of standard ULDs. Each ULD is characterized by: - a pickup point (the FF), - a destination point (the intended GH), - a pickup time window (a time interval where the ULD can be picked up), - a delivery time window (a time interval where the ULD can be delivered), - a weight, and - a lateral occupancy. Time windows are imposed on both the pickup and the delivery, generally for different reasons. For example, the ULD might not be ready for pickup before a specific time, which makes the lower (early) bound of the pickup time window an important piece of information to avoid unnecessary waiting times. Conversely, for deliveries the upper (late) bound of the time window is of paramount importance to make sure the ULD is loaded on time onto the allocated flight and to avoid additional rebooking costs; see Liu et al. (2019) for an estimation of additional airline costs due to late package delivery. Finally, weight and lateral occupancy are also considered, since trucks are capacitated in both weight and size.

The transportation setting consists of trucks departing from a central depot, visiting first FFs, to pick up ULDs, then GHs for the delivery, and finally returning to the initial depot. Since the ULDs have a width comparable to the one of the trucks, they have to be discharged at the GHs according to a LIFO strategy. Contrary to FFs, we assume that each GH has a subset of export docks reserved

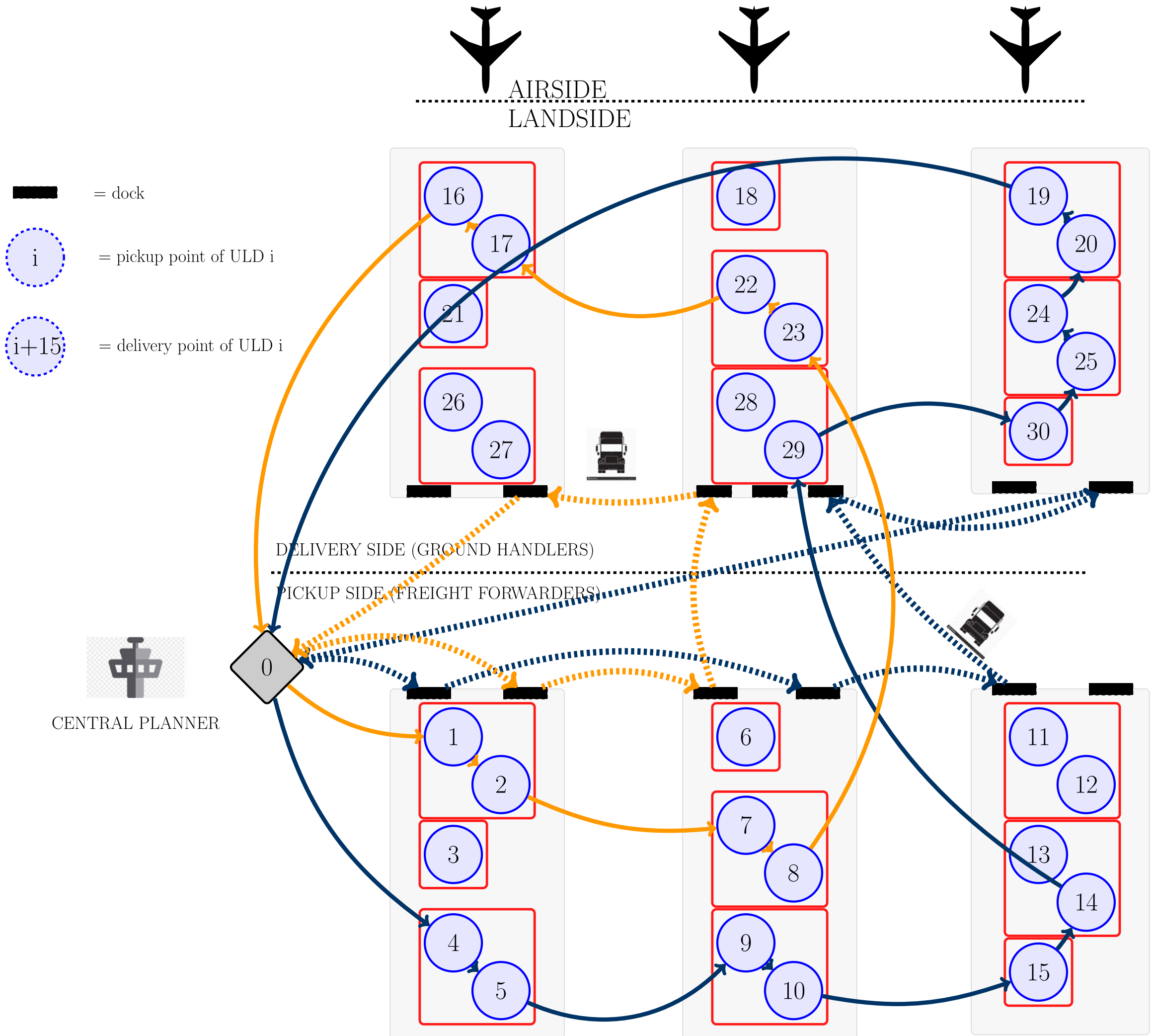

Fig. 3. Example of two routes, involving a subset of nodes, as sequences of pickup and delivery nodes (solid line) and as sequences of warehouses visited (dashed lines). The same color represents the same route using the two different perspectives. The routes start from and return to the central depot 0.. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

for the consortium; hence, there is a maximum number of trucks that can be processed simultaneously. This is reasonable since even major cargo airports are generally served only by a handful of GHs (given the necessity to have their warehouses overlooking runways), which serve hundreds of different forwarding companies. Consequently, bottleneck issues and truck queuing are much more common outside GHs rather than FFs.

The goal of the central planner is to minimize waiting times at the GHs and routing times, while making sure that: all ULDs are transported within their time windows; capacity restrictions are met, and that FFs and GHs are visited at most once in a trip. Waiting times may arise on two different levels. First, when trucks wait for a free dock at the GHs' side. Second, due to ULDs' time windows, docked trucks may need to wait for the opening time of an ULD. In addition, in line with current pilot programs such as eLink (ACN, 2021), we assume that all paper-based documentation is replaced with digital documentation. As such, truckers headed to a GH do not need to park and provide paperwork at the GH office, but can directly go to the docking station compatibly with the instructions of the central planner. See Fig. 3 for a graphical representation of the problem.

3.2. GHDC-PDPTW: general notation

For the mathematical notations, unless specified otherwise, we will be using a calligraphic font for sets, uppercase letters for parameters, and lowercase letters for decision variables.

With reference to cargo, we define the set $\mathcal{U}$ of ULDs. Binary parameters F_{uj} and G_{uj} indicate respectively whether ULD u is related to FF j and GH j. Let W_u and L_u be respectively the weight and the length of ULD $u \in \mathcal{U}$. Pairs $[E_u^{FF}; D_u^{FF}]$ and $[E_u^{GH}; D_u^{GH}]$

are the time windows of ULD u respectively at the related FF and GH. Finally, P_u is the needed handling time of ULD u for loading and unloading. Concerning the fleet, $\mathcal{K}$ is the set of homogeneous trucks, featuring weight capacity Q^W and side length Q^L. The overall set of nodes is defined as follows. We define the set of FFs' warehouses as $\mathcal{F}$ (where $\mathcal{F}_f$ is the fth FF in the set), the set of GHs' warehouses as $\mathcal{G}$ (where $\mathcal{G}_g$ is the gth GH in the set). The depot, node 0, is the location where trucks both start and end their tour. T_{ij} is the transport time between nodes i and j. Finally, Δ_{GH} is the number of docks at the GHs. For simplicity, we assume the same number for each GH. The extension to the case where docks are GH-specific is straight-forward and entails the definition of Δ^g_{GH} as the number of docks available at GH g. In our case study the different GHs are comparable in terms of size, therefore we opted for the same value Δ_{GH}.

In the following two subsections we provide the description of the two models, pointing out the adapted sets and parameters, and the relevant variables.

3.3. GHDC-PDPTW: a routing-based formulation

In the first model, henceforth M_1, routing is carried out at the ULD level. In particular, for each ULD $u \in \mathcal{U}$ we define two nodes, a pickup node on the FF side and a delivery node on the GH side. We define the sets of pickup and delivery nodes $\mathcal{N}_P$ and $\mathcal{N}_D$, respectively. The cardinality of both sets is $|\mathcal{U}| = n$, i.e., the number of ULDs considered. Notation-wise, the central depot is node 0 (and serves as both the origin and the destination depot in our model, but the extension to the case where they are distinct is straight-forward), nodes belonging to $\mathcal{N}_P$ range from 1 to n, and nodes belonging to $\mathcal{N}_D$ range from $n+1$ to $2n$. It also follows that for every pickup node $i \in \mathcal{N}_P$, the associated delivery node is $i+n \in \mathcal{N}_D$. Nodes in $\mathcal{N}_P$ and $\mathcal{N}_D$ inherit the position of the warehouse they belong to. M_1 is based on a directed graph $\mathcal{G}_1 = (\mathcal{N}_1, \mathcal{E}_1)$ where $\mathcal{N}_1$ is the set of nodes and $\mathcal{E}_1$ is the set of edges. $\mathcal{N}_1$ is the union of 3 node sets: $\mathcal{N}_1 = 0 \cup \mathcal{N}_P \cup \mathcal{N}_D$ with $|\mathcal{N}_1| = 2n+1$. $\mathcal{E}_1$ contains all the feasible edges in the graph, taking into account that FFs should be visited before GHs during a tour and LIFO restrictions. Because of LIFO restrictions, for every $i \in \mathcal{N}_P$ the only edge towards $\mathcal{N}_D$ is the one connecting i with $i+n$. Having introduced sets $\mathcal{N}_P$ and $\mathcal{N}_D$, we re-define time windows as $[E_i; D_i]$ by dropping the FF superscript, that would characterize nodes $i \in \mathcal{N}_P$, and the GH superscript, that would characterize nodes $i \in \mathcal{N}_D$.

The description of the decision variables follows. x^k_{ij} is a binary decision variable that is unitary if truck k moves from node i to node j. Together with this main routing variable, M_1 includes continuous time-related decision variables. τ_i represents the start of service time for every node $i \in \mathcal{N}_P \cup \mathcal{N}_D$; τ^k_0/τ^k_{end} define the start/end of service time of truck k, a^F_{kf}/d^F_{kf} define the arrival/departure time of truck k to/from FF f, a^G_{kg}/d^G_{kg} define the arrival/departure time of truck k to/from GH g. w^D_{kg} computes the waiting time of truck k at GH g caused by the unavailability of a dock, while w^F_{kf} and w^G_{kg} define, respectively, the waiting times of truck k at FF f and GH g while docked. Finally, binary decision variables η, y, and z are used to assign trucks to docks in each GH. $\eta^g_{k_1,k_2}$ is a binary variable that is unitary if trucks k_1 and k_2 both visit GH g without overlapping schedules. y^g_{kd} is a binary variable that equals 1 if truck k is assigned to dock d when visiting GH g. $z^g_{k_1,d_1,k_2,d_2}$ is a binary variable used to linearize the product $y^g_{k_1,d_1} \cdot y^g_{k_2,d_2}$. A list of all sets, variables and parameters of M_1 is reported in Table 1.

We formulate M_1 as follows:

$$min \sum_{k\in\mathcal{K}} \sum_{(i,j)\in\mathcal{E}_1} T_{ij} x^k_{ij} + \left(\sum_{k\in\mathcal{K}} \sum_{g\in\mathcal{G}} w^D_{kg} + \sum_{k\in\mathcal{K}} \sum_{g\in\mathcal{G}} w^G_{kg} + \sum_{k\in\mathcal{K}} \sum_{f\in\mathcal{F}} w^F_{kf} \right) \tag{1}$$

subject to:

$$\sum_{k\in\mathcal{K}} \sum_{(j,i)\in\mathcal{E}_1} x^k_{ji} = 1 \qquad \forall i \in \mathcal{N}_P \tag{2}$$

$$\sum_{(j,i)\in\mathcal{E}_1} x^k_{ji} - \sum_{(j,i+n)\in\mathcal{E}_1} x^k_{j,i+n} = 0 \qquad \forall i \in \mathcal{N}_P, \forall k \in \mathcal{K} \tag{3}$$

$$\sum_{(0,i)\in\mathcal{E}_1} x^k_{0i} \le 1 \qquad \forall k \in \mathcal{K} \tag{4}$$

$$\sum_{(j,i)\in\mathcal{E}_1} x^k_{ji} - \sum_{(i,j)\in\mathcal{E}_1} x^k_{ij} = 0 \qquad \forall i \in \mathcal{N}_1, \forall k \in \mathcal{K} \tag{5}$$

$$x^k_{0i} - x^k_{i+n,0} = 0 \qquad \forall i \in \mathcal{N}_P, \forall k \in \mathcal{K} \tag{6}$$

$$x^k_{ij} - x^k_{j+n,i+n} = 0 \qquad \forall (i,j) \in \mathcal{E}_1 : 1 \le i, j \le n, \forall k \in \mathcal{K} \tag{7}$$

$$\sum_{i\in\mathcal{F}_f} \sum_{j\in\mathcal{F}\setminus\{\mathcal{F}_f\}} x^k_{ji} + \sum_{i\in\mathcal{F}_f} x^k_{0,i} \le 1 \qquad \forall f \in \mathcal{F}, \forall k \in \mathcal{K} \tag{8}$$

$$\sum_{i\in\mathcal{G}_g} \sum_{j\in\mathcal{G}\setminus\{\mathcal{G}_g\}} x^k_{ji} + \sum_{i\in\mathcal{G}_g} x^k_{i-n,i} \le 1 \qquad \forall g \in \mathcal{G}, \forall k \in \mathcal{K} \tag{9}$$

$$\tau_j \ge \tau_i + P_i + T_{ij} - (1 - \sum_{k\in\mathcal{K}} x^k_{ij})M \qquad \forall (i,j) \in \mathcal{E}_1 : 1 \le j \le n \tag{10}$$

$$\tau_j \ge \tau_i + P_i + T_{ij} - (1 - x^k_{ij})M + w^D_{kg} \qquad \forall g \in \mathcal{G}, (i,j) \in \mathcal{E}_1 : n+1 \le j \le 2n \wedge (j \in \mathcal{G}_g \wedge i \notin \mathcal{G}_g), \forall k \in \mathcal{K} \tag{11}$$

Table 1
Sets, parameters, and variables of M_1.

Sets	
$\mathcal{N}_P$	Set of pickup nodes
$\mathcal{N}_D$	Set of delivery nodes
$\mathcal{N}_1$	Set of all nodes, index 0 for depot
$\mathcal{F}$	Set of FFs
$\mathcal{G}$	Set of GHs
$\mathcal{E}_1$	Set of all edges
$\mathcal{K}$	Set of trucks
Parameters	
T_{ij}	Travel time between nodes i and j
Δ_{GH}	Number of docks at the GHs
W_i	Weight of the ULD associated to node i
L_i	Length of the ULD associated to node i
E_i	Release time of the ULD associated to node i
D_i	Due date of the ULD associated to node i
P_i	Processing time of the ULD associated to node i
Q^W	Weight capacity of a truck
Q^L	Length of the trailer of a truck
Variables	
x_{ij}^k	Binary variable, equals 1 if truck k moves from node i to node j
τ_0^k, τ_{end}^k	Start/End of service time of truck k
τ_i	Start of service time at node i
a_{kf}^F/d_{kf}^F	Arrival/Departure time of truck k to/from FF f
a_{kf}^G/d_{kf}^G	Arrival/Departure time of truck k to/from GH g
w_{kg}^D	Waiting time of truck k at GH g before docking
w_{kf}^F	Waiting time of truck k at FF f while docked
w_{kg}^G	Waiting time of truck k at GH g while docked
η_{k_1,k_2}^g	Binary variable, equals 1 if the departure time of truck k_1 from ground handler g is smaller or equal to the docking time of truck k_2 to ground handler g
y_{kd}^g	Binary variable, equals 1 if truck k is assigned to dock d in ground handler g
z_{k_1,d_1,k_2,d_2}^g	Binary variable, equals 1 if truck k_1 is assigned to dock d_1 and truck k_2 is assigned to dock d_2 in ground handler g

$$\tau_j \geq \tau_i + P_i + T_{ij} - (1 - \sum_{k\in\mathcal{K}} x_{ij}^k)M \qquad \forall g \in \mathcal{G}, (i,j) \in \mathcal{E}_1 : n+1 \leq j \leq 2n \wedge (i,j) \in \mathcal{G}_g \tag{12}$$

$$\tau_{end}^k \geq \tau_i + P_i + T_{ij} - (1 - x_{ij}^k)M \qquad \forall (i,0) \in \mathcal{E}_1, \forall k \in \mathcal{K} \tag{13}$$

$$E_i \leq \tau_i \leq D_i \qquad \forall i \in \mathcal{N}_P \cup \mathcal{N}_D \tag{14}$$

$$\sum_{(j,i)\in\mathcal{E}_1 : i\leq n} W_i x_{ji}^k \leq Q^W \qquad \forall k \in \mathcal{K} \tag{15}$$

$$\sum_{(j,i)\in\mathcal{E}_1 : i\leq n} L_i x_{ji}^k \leq Q^L \qquad \forall k \in \mathcal{K} \tag{16}$$

$$a_{kf}^F \geq \tau_i + P_i + T_{ij} - (1 - x_{ij}^k)M \qquad \forall f \in \mathcal{F}, \forall (i,j) \in \mathcal{E}_1 : j \in \mathcal{F}_f \wedge i \notin \mathcal{F}_f, \forall k \in \mathcal{K} \tag{17}$$

$$a_{kf}^F \leq \tau_i + P_i + T_{ij} + (1 - x_{ij}^k)M \qquad \forall f \in \mathcal{F}, \forall (i,j) \in \mathcal{E}_1 : j \in \mathcal{F}_f \wedge i \notin \mathcal{F}_f, \forall k \in \mathcal{K} \tag{18}$$

$$d_{kf}^F \geq \tau_i + P_i - (1 - x_{ij}^k)M \qquad \forall f \in \mathcal{F}, \forall (i,j) \in \mathcal{E}_1 : i \in \mathcal{F}_f \wedge j \notin \mathcal{F}_f, \forall k \in \mathcal{K} \tag{19}$$

$$d_{kf}^F \leq \tau_i + P_i + (1 - x_{ij}^k)M \qquad \forall f \in \mathcal{F}, \forall (i,j) \in \mathcal{E}_1 : i \in \mathcal{F}_f \wedge j \notin \mathcal{F}_f, \forall k \in \mathcal{K} \tag{20}$$

$$a_{kg}^G \geq \tau_i + P_i + T_{ij} - (1 - x_{ij}^k)M \qquad \forall g \in \mathcal{G}, \forall (i,j) \in \mathcal{E}_1 : j \in \mathcal{G}_g \wedge i \notin \mathcal{G}_g, \forall k \in \mathcal{K} \tag{21}$$

$$a_{kg}^G \leq \tau_i + P_i + T_{ij} + (1 - x_{ij}^k)M \qquad \forall g \in \mathcal{G}, \forall (i,j) \in \mathcal{E}_1 : j \in \mathcal{G}_g \wedge i \notin \mathcal{G}_g, \forall k \in \mathcal{K} \tag{22}$$

$$d_{kg}^G \geq \tau_i + P_i - (1 - x_{ij}^k)M \qquad \forall g \in \mathcal{G}, \forall (i,j) \in \mathcal{E}_1 : i \in \mathcal{G}_g \wedge j \notin \mathcal{G}_g, \forall k \in \mathcal{K} \tag{23}$$

$$d_{kg}^G \leq \tau_i + P_i + (1 - x_{ij}^k)M \qquad \forall g \in \mathcal{G}, \forall (i,j) \in \mathcal{E}_1 : i \in \mathcal{G}_g \wedge j \notin \mathcal{G}_g, \forall k \in \mathcal{K} \tag{24}$$

$$w_{kf}^F \geq d_{kf}^F - a_{kf}^F - \sum_{i\in\mathcal{F}_f} \sum_{(j,i)\in\mathcal{E}_1} P_i x_{ji}^k \qquad f = 1, \dots, |\mathcal{F}|, \forall k \in \mathcal{K} \tag{25}$$

$$w_{kg}^G \geq d_{kg}^G - a_{kg}^G - w_{kg}^G - \sum_{i\in\mathcal{G}_g} \sum_{(j,i)\in\mathcal{E}_1} P_i x_{ji}^k \qquad g = 1, \dots, |\mathcal{G}|, \forall k \in \mathcal{K} \tag{26}$$

$$d^G_{k_1,g} - a^G_{k_2,g} - w^D_{k_2^g} + M\eta^g_{k_1,k_2} \leq M \qquad g = 1,\dots,|\mathcal{G}|, \forall (k_1,k_2) \in \mathcal{K}\times\mathcal{K} \tag{27}$$

$$-d^G_{k_1,g} + a^G_{k_2,g} + w^D_{k_2,g} - M\eta^g_{k_1,k_2} \leq 0 \qquad g = 1,\dots,|\mathcal{G}|, \forall (k_1,k_2) \in \mathcal{K}\times\mathcal{K} \tag{28}$$

$$\sum_{d=1}^{\Delta_{GH}} y^g_{kd} = \sum_{i\in\mathcal{G}_g}\sum_{j\in\mathcal{G}\setminus\{\mathcal{G}_g\}} x^k_{ji} + \sum_{i\in\mathcal{G}_g} x^k_{i-n,i} \qquad g = 1,\dots,|\mathcal{G}|, \forall k \in \mathcal{K} \tag{29}$$

$$z^g_{k_1,d_1,k_2,d_2} \leq y^g_{k_1,d_1} \qquad \forall (k_1,k_2) \in \mathcal{K}\times\mathcal{K} : k_1 < k_2, d_1, d_2 = 1,\dots,\Delta_{GH}, g = 1,\dots,|\mathcal{G}| \tag{30}$$

$$z^g_{k_1,d_1,k_2,d_2} \leq y^g_{k_2,d_2} \qquad \forall (k_1,k_2) \in \mathcal{K}\times\mathcal{K} : k_1 < k_2, d_1, d_2 = 1,\dots,\Delta_{GH}, g = 1,\dots,|\mathcal{G}| \tag{31}$$

$$y^g_{k_i,d_1} + y^g_{k_2,d_2} - 1 \leq z^g_{k_1,d_1,k_2,d_2} \qquad \forall (k_1,k_2) \in \mathcal{K}\times\mathcal{K} : k_1 < k_2, d_1, d_2 = 1,\dots,\Delta_{GH}, g = 1,\dots,|\mathcal{G}| \tag{32}$$

$$z^g_{k_1,d_1,k_2,d_1} \leq \eta^g_{k_1,k_2} + \eta^g_{k_2,k_1} \qquad \forall (k_1,k_2) \in \mathcal{K}\times\mathcal{K} : k_1 < k_2, d_1 = 1,\dots,\Delta_{GH}, g = 1,\dots,|\mathcal{G}| \tag{33}$$

$$x^k_{ij}, \eta^g_{k_1,k_2}, y^g_{kd} \qquad \in \{0,1\} \tag{34}$$

$$\tau^k_0, \tau^k_{end}, a^F_{kf}, d^F_{kf}, a^G_{kg}, d^G_{kg}, w^D_{kg}, w^G_{kg}, w^F_{kf}\, z^g_{k_1,d_1,k_2,d_2} \qquad \in \mathbf{R}^+ \tag{35}$$

Objective function (1) defines the overall transportation time. The first term accounts for traveling times between nodes. The second term defines the waiting times of each truck, respectively at: GHs before docking, GHs while unloading ULDs, and at FFs while loading ULDs.

Constraint (2) ensures that every ULD is picked up. Constraint (3) imposes that a $(i, i+n)$ pickup-delivery pair is visited by the same truck. Inequality (4) enforces that every truck is used at most once. Flow conservation is imposed by constraint (5). Equalities (6) and (7) model the LIFO strategy. With (6) if i is the first pickup node visited, then its delivery counterpart must be the last one. (7) imposes that a sequence of two pickup nodes corresponds to the reverse sequence of their associated delivery nodes. Constraints (8) and (9) ensure that every truck visits each FF and GH, respectively, at most once. Note that a flow conservation set of constraints per warehouse is not necessary, because flow conservation at the warehouse level is automatically satisfied thanks to flow conservation at the ULD level (equality (5)).

Constraint (10) models time precedence constraints for pickup nodes. Since time decision variable τ_i is not truck-specific (the truck visiting node i is easily identifiable via constraint (2)), inside the brackets the summation over all trucks of decision variables x_{ij} is carried out. (11) and (12) enforce time precedence constraints to delivery nodes. Constraint (11) models time precedence between a node i not belonging to GH g and a node j belonging to GH g. The start of service time at node j is the arrival time at that GH plus a potential waiting time. On the other hand, there is no waiting time between nodes belonging to the GH (constraint (12)). Constraint (13) enforces time precedence between delivery nodes and the destination depot.

Constraint (14) imposes time windows on pickup and delivery nodes. (15) and (16) enforce, respectively, weight and length capacity for each truck. Due to the assumption that all pickups precede all deliveries in a truck tour, constraints are not imposed for every node pair, but as a single constraint per truck that encompasses every possible pickup node. Inequalities (17) and (18) define the arrival time of a truck at a FF as the start of service time of the previous node (not belonging to the FF) plus its processing time and the traveling time to the FF. Inequalities (19) and (20) define the departure time of a truck from a FF as the start of service time of the last node served in that FF plus its processing time. Constraints (21) to (24) define the same variables for GHs. Inequality (25) enforces that the waiting time of a truck at a FF while loading ULDs is greater or equal to the departure time from that FF minus the arrival time to that FF minus the overall processing time of the ULDs served by the truck in that FF. Constraint (26) applies the same concept to GHs. Here, we also subtract the waiting time the truck might experience before docking, since the actual docking time is $aG^g_k + w^g_k$. For FFs, where docking capacity is unlimited, we have that docking time always coincides with the arrival time. Constraints (27) and (28) define the binary value of the overlap variable $\eta^g_{k_i,k_j}$ with respect to the undocking (departure) time of truck k_i and the docking (arrival plus waiting) time of truck k_j.

Constraint (29) ensures that if a truck visits a GH, then it must be assigned to a dock. Constraints from (30) to (32) enforce that decision variable $z^g_{k_1,d_1,k_2,d_2}$ is 1 if and only if both $y^g_{k_1,d_1}$ and $y^g_{k_2,d_2}$ are 1 and 0 otherwise. Constraint (33) ensures that two trucks can be assigned to the same dock if and only if their time schedules do not overlap. Note that, while we cycle over every combination of docks (considering the case where the two trucks are assigned to the same dock as well), we enforce $k_1 < k_2$ to avoid unnecessary and redundant assignment decision variables. Finally, constraints (34) and (35) define the nature of the decision variables. In particular, it should be noted that decision variable $z^g_{k_1,d_1,k_2,d_2}$ is relaxed to be continuous in the $[0,1]$ interval, although being in essence a binary variable. This is possible thanks to constraint (32) that forces the associated decision variable to be either zero or one.

3.4. GHDC-PDPTW: a bin packing-based formulation

In the second model, henceforth M_2, we focus on allocating the ULDs to the trucks and based on this allocation we design the routing. Thereby, we can reduce the size of the network substantially and consider each node as either a FF or GH.

Similarly to M_1, M_2 is based on a directed graph $\mathcal{G}_2 = (\mathcal{N}_2, \mathcal{E}_2)$ where $\mathcal{N}_2$ is the set of nodes and $\mathcal{E}_2$ the set of edges. Since nodes correspond to warehouses in M_2, we have $|\mathcal{N}_2| = |\mathcal{F}| + |\mathcal{G}| + 1$. Generally speaking, the number of warehouses in this problem is smaller than the number of ULDs considered. Hence, $|\mathcal{F}| + |\mathcal{G}| + 1 \ll 2n + 1 \implies |\mathcal{N}_2| \ll |\mathcal{N}_1|$. The information on the ULDs is

Table 2
Sets, parameters, and variables of M_2.

Sets	
$\mathcal{U}$	Set of ULD
$\mathcal{F}$	Set of FF
$\mathcal{G}$	Set of GH
$\mathcal{N}_2$	Set of all nodes, index 0 for depot
$\mathcal{K}$	Set of trucks
Parameters	
T_{ij}	Travel time between terminals i and j
Δ_{GH}	Number of docks at the GHs
F_{uj}	1 if ULD u is originated at FF j, 0 otherwise
G_{uj}	1 if ULD u is destined to GH j, 0 otherwise
W_u	Weight of ULD u
L_u	Length of ULD u
E_u^{FF}, E_u^{GH}	Earliest start time of ULD u, respectively at the related FF and GH
D_u^{FF}, D_u^{GH}	Due date of ULD u, respectively at the related FF and GH
P_u	Processing time of ULD u
Q^W	Weight capacity of a truck
Q^L	Length of a truck
Variables	
f_{uk}	Binary variable, equals 1 if ULD u is allocated to truck k
x_{ij}^k	Binary variable, equals 1 if truck k travels from i to j
τ_u^{FF}	Start of service time of ULD u on the FF side
τ_u^{GH}	Start of service time of ULD u on the GH side
t_i^k	Time truck k reaches node i
t_0^k	Departure time of truck k
p_j^k	Total processing time of truck k at node j
$y_{u'u''}^{FF}$	Precedence binary variable between ULDs u' and u'' at the FF
z_{k_1,k_2}^g	Precedence binary variable between trucks k_1 and k_2 at GH g
h_{gj}^k	Binary variable, equals 1 if truck k is allocated to dock j of GH g
p_i^k	Time spent by truck k at node i
w_i^k	Waiting time accumulated by truck k at either GH or FF i

considered in a binary packing variable f_{uk}, indicating whether ULD u is allocated to truck k. Therefore, all constraints that need to take into account the individual container features, such as weight, time, etc., will be related to f_{uk}.

Before presenting the formulation, we revise some of the previously defined sets and parameters. With respect to the decision variables, we redefine a few of them. Variable t_i^k indicates the time truck k reaches node i, whereas τ_u^{FH} and τ_u^{GH} are respectively the starting times ULD u is processed at the FF and GH. Note that the superscripts FH and GH are not indexes. Finally, the new variables, besides f_{uk}, are $y_{u'u''}^{FF}$, $y_{k'k''}^{i}$, p_i^k, and w_i^k. Binary variable $y_{u'u''}^{FF}$ indicates, if 1, that ULD u' is processed before u'', and is 0 otherwise. $y_{k'k''}^{i}$ is a binary variable which takes value 1 if trucks k' is processed before k'' at GH i. Variable p_i^k computes the time spent by truck k at node i. Finally, w_i^k computes the waiting time accumulated by truck k at node $i \in \mathcal{F} \cup \mathcal{G}$. A list of all sets, variables, and parameters of M_2 is reported in Table 2.

We formulate M_2 as follows:

$$min \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{N}_2} \sum_{j\in\mathcal{N}_2} T_{ij} x_{ij}^k + \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{N}_2\setminus\{0\}} w_i^k \tag{36}$$

subject to:

$$\sum_{k\in\mathcal{K}} f_{uk} = 1 \qquad \forall u \in \mathcal{U} \tag{37}$$

$$\sum_{u\in\mathcal{U}} W_u f_{uk} \le Q^W \qquad \forall k \in \mathcal{K} \tag{38}$$

$$\sum_{u\in\mathcal{U}} L_u f_{uk} \le Q^L \qquad \forall k \in \mathcal{K} \tag{39}$$

$$\sum_{j\in\mathcal{N}_2\setminus\{0\}} x_{0,j}^k \le 1 \qquad \forall k \in \mathcal{K} \tag{40}$$

$$\sum_{j\in\mathcal{N}_2:j\neq i} x_{ij}^k - \sum_{j\in\mathcal{N}_2:j\neq i} x_{ji}^k = 0 \qquad \forall i \in \mathcal{N}_2, \forall k \in \mathcal{K} \tag{41}$$

$$f_{uk} \le \sum_{i\in\mathcal{N}_2\setminus\{j\}} x_{ij}^k \qquad \forall k \in \mathcal{K}, \forall u \in \mathcal{U}, j \in \mathcal{G} : G_{uj} = 1 \tag{42}$$

$$f_{uk} \leq \sum_{i \in \mathcal{F} \setminus \{j\} \cup \{0\}} x_{ij}^k \qquad \forall k \in \mathcal{K}, \forall u \in \mathcal{U}, j \in \mathcal{F} : F_{uj} = 1 \tag{43}$$

$$t_j^k \geq t_i^k + p_i^k + T_{ij} - (1 - x_{ij}^k)M \qquad \forall (i,j) \in \mathcal{N}_2 : j \neq 0, \forall k \in \mathcal{K} \tag{44}$$

$$w_i^k \geq p_i^k - \sum_{u \in \mathcal{U} : u \in \mathcal{G}_i} f_{uk} P_u \qquad \forall i \in \mathcal{G}, \forall k \in \mathcal{K} \tag{45}$$

$$w_i^k \geq p_i^k - \sum_{u \in \mathcal{U} : u \in \mathcal{F}_i} f_{uk} P_u \qquad \forall i \in \mathcal{F}, \forall k \in \mathcal{K} \tag{46}$$

$$\tau_i^{FF} + P_i \leq \tau_j^{FF} + (3 - y_{u'u''}^{FF} - f_{u'k} - f_{u''k})M \qquad \forall u', u'' \in \mathcal{U} : u' < u'', k \in \mathcal{K} \tag{47}$$

$$\tau_j^{FF} + P_j \leq \tau_i^{FF} + (2 + y_{ij}^{FF} - f_{ik} - f_{jk})M \qquad \forall i, j \in \mathcal{U} : i < j, k \in \mathcal{K} \tag{48}$$

$$E_u^{FF} \leq \tau_u^{FF} \leq D_u^{FF} \qquad \forall u \in \mathcal{U} \tag{49}$$

$$\tau_u^{FF} \geq t_j^k - (1 - f_{uk})M \qquad \forall u \in \mathcal{U}, \forall k \in \mathcal{K}, j \in \mathcal{F} : F_{uj} = 1 \tag{50}$$

$$p_j^k \geq \tau_u^{FF} + P_u - t_j^k - (1 - f_{uk})M \qquad \forall u \in \mathcal{U}, k \in \mathcal{K}, j \in \mathcal{F} : F_{uj} = 1 \tag{51}$$

$$\tau_i^{GH} + P_{u'} \leq \tau_j^{GH} + (2 + y_{u'u''}^{FF} - f_{u'k} - f_{u''k})M \qquad \forall u', u'' \in \mathcal{U} : u' < u'', \forall k \in \mathcal{K} \tag{52}$$

$$\tau_{u''}^{GH} + P_{u''} \leq \tau_{u'}^{GH} + (3 - y_{u'u''}^{FF} - f_{u'k} - f_{u''k})M \qquad \forall u', u'' \in \mathcal{U} : u' < u'', \forall k \in \mathcal{K} \tag{53}$$

$$E_u^{GH} \leq \tau_u^{GH} \leq L_u^{GH} \qquad \forall u \in \mathcal{U} \tag{54}$$

$$\tau_u^{GH} \geq t_j^k - (1 - f_{uk})M \qquad \forall u \in \mathcal{U}, \forall k \in \mathcal{K}, j \in \mathcal{G} : G_{uj} = 1 \tag{55}$$

$$p_j^k \geq \tau_u^k + P_u - t_j^k - (1 - f_{uk})M \qquad \forall u \in \mathcal{U}, \forall k \in \mathcal{K}, j \in \mathcal{G} : G_{uj} = 1 \tag{56}$$

$$\sum_{j=1}^{\Delta_{GH}} h_{ij}^k = \sum_{j=1}^{|\mathcal{N}_2|} x_{ji}^k \qquad \forall k \in \mathcal{K}, i \in \mathcal{G} \tag{57}$$

$$t_i^{k_1} + p_i^{k_1} \leq t_i^{k_2} + (2 + z_{k_1,k_2}^i - h_{ij}^{k_1} - h_{ij}^{k_2})M \qquad \forall i \in \mathcal{G}, j = 1 \dots \Delta_{GH}, \forall (k_1, k_2) \in \mathcal{K} \times \mathcal{K} : k_1 < k_2 \tag{58}$$

$$t_i^{k2} + p_i^{k_2} \leq t_i^{k_1} + (3 - z_{k_1,k_2}^i - y_{ij}^{k_1} - y_{ij}^{k_2})M \qquad \forall i \in \mathcal{G}, j = 1 \dots \Delta_{GH}, \forall (k_1, k_2) \in \mathcal{K} \times \mathcal{K} : k_1 < k_2 \tag{59}$$

$$f_{uk}, x_{ij}^k, y_{u'u''}^{FF}, h_{ij}^k, z_{k_1,k_2}^i \in \{0, 1\} \tag{60}$$

$$t_i^k, p_i^k, w_k^i, \tau_u^{FF}, \tau_u^{GH} \in \mathbf{R}^+ \tag{61}$$

Objective function (36) minimizes the sum of transport and waiting times of the fleet. Equality (37) imposes that every ULD is assigned to a truck, whereas constraints (38) and (39) impose, respectively, weight and length capacity on each truck. Inequality (40) enforces that each truck is used at most once. Constraint (41) ensures conservation of flow for each node $\in \mathcal{N}_2$. Inequality (42) states that if a ULD is assigned to a truck, then the truck must visit the GH the ULD is associated to. Constraint (43) applies the same rationale to the FF side. Constraint (44) enforces time precedence between nodes for each truck's route. With inequalities (45) and (46) we compute the waiting time at the GHs' and FFs' warehouses respectively. Constraints (47) and (48) define time precedence constraints between ULDs allocated to the same truck and constraint (49) imposes time windows on ULDs on the FF side. Inequality (50) ensures that the start of service time of a ULD in the associated FF should be greater of equal than the arrival time of the intended truck at that specific FF. Constraints (51) compute the total processing time of a truck at a FF, defined as the latest service time of an ULD plus its task time. Constraints (52)–(56) are the equivalent form of constraints (47)–(51) for GHs. Note that in constraints (47)–(48) and (52)–(53) the expressions inside the brackets are reversed to ensure a LIFO loading. With (54) we impose the time windows for each ULD. Inequalities (55) and (56) are the equivalent of (50) and (51) at the GHs. From (57) to (59) we avoid overlapping between trucks if docked at the same GH dock. In particular, with (57) we allocate each truck to a dock, and with (58) and (59) we define the precedence. Finally, (60) and (61) define the nature of the decision variables.

4. A metaheuristic approach

Along with the two proposed formulations, we also develop a metaheuristic approach based on an ALSN framework to modify the current routing and scheduling solution and on Simulated Annealing (SA) to diversify the search in the solution space. The framework is inspired by Ropke and Pisinger (2006), although some ad-hoc operators are designed to consider the unique features of the GHDC-PDPTW. We first describe how we generate an initial solution. Next the main algorithm is presented.

4.1. Initial solution generation

The first step of our solution method is the generation of either a feasible or a weakly infeasible initial solution $\mathcal{S}_0$. The latter is composed of routes that are individually feasible, but taken together they either violate GH dock capacity and or do not include of all ULDs. This will be the only type of infeasible solution accepted by our algorithm, since these solutions can potentially become feasible either by just tweaking trucks departure times or by properly inserting unassigned ULDs.

Additionally, we evaluate the quality of a solution and the level of infeasibility through the following cost function:

$$\mathcal{J}(\mathcal{S}) = \sum_{r=1}^{|\mathcal{R}|}(t^r_{end} - t^r_0) + \alpha \mathbb{W}_{dc}(\mathcal{S}) + \beta |\mathcal{C}_{\Downarrow}|(\mathcal{S}) \tag{62}$$

where the first term identifies the overall traveling time ($|\mathcal{R}|$ is the number of trucks used in $\mathcal{S}$), the second term penalizes the overall dock capacity violation $\mathbb{W}_{dc}$ (expressed in time units) incurred by $\mathcal{S}$, and the third term penalizes ULDs that are not delivered in the current solution. In this regard, $\mathcal{C}_{\Downarrow}$ is the set of unassigned ULDs, as opposed to $\mathcal{C}_{\Uparrow}$. α and β are parameters used to leverage the effects of these two costs.

Given our set of ULDs $\mathcal{U}$, to construct $\mathcal{S}_0$, we rely on a basic greedy insertion operator as follows. We initialize an empty route k and insert one by one ULDs that cause the least additional cost to $\mathcal{J}_k$. Note that at this stage we still do not consider the second and third cost items in Eq. (62), as these will be computed at the end of the allocation of all ULDs. The process is repeated with the remaining unassigned ULDs until no feasible insertion is possible. Note that, while the position of the first insertion is fixed and results in the partial route $\{0, i, i+n, 2n+1\}$, for every other ULD a number of insertion points equal to $n_r + 1$ (where n_r is the number of ULDs currently in route r) need to be checked. The selected ULD will be placed in the minimum cost position.

After the initial set of feasible routes is created, we need to check if the overall solution is also feasible or weakly infeasible due to dock capacity violations. To do so, we have developed an ad-hoc routine that detects dock capacity violations of the current solution $\mathcal{S}$ and attempts to remove them sequentially. If at least one dock capacity violation is found, all occurrences are stored in chronological order as we attempt to solve them sequentially. We determine which truck can be delayed the smallest amount of time to solve the violation. The additional delay can be even zero if the truck's departure can simply be delayed without compromising due dates and other schedules upstream. We keep updating schedules and solving the first violation until either no violations are found, or the first violation cannot be feasibly solved.

4.2. The ALNS algorithm

After the creation of the initial solution, the algorithm randomly selects removal or insertion operators to modify the current solution to generate a new solution $\mathcal{S}'$. The probability of selection is proportional to the success rate of the operators. Next, the solution is accepted based on the Boltzmann Distribution, whose parameter T (temperature) is decreased based on the SA principle.

About removal operators, they partially destroy the current solution $\mathcal{S}$ by removing a number of $\mathbf{q}$ ULDs. This number should be set reasonably high to increase the chances of finding better solutions in a single iteration, at the cost of a slower computation.

We rely on the following removal operators:

- random removal: $\mathbf{q}$ ULDs are randomly removed from $\mathcal{S}$.
- Shaw removal (Shaw, 1997): ULDs are removed from a truck using a similarity measure, i.e., a relatedness measure $R(i, j)$ that maps how similar ULDs i and j are. The goal of this operator is to remove ULDs that share some similarities, since it should be easier to re-shuffle them and insert them in different trucks.
- best removal: for every ULD $i \in \mathcal{C}_{\Uparrow}$, we compute $\Delta\mathcal{J}(i) = \mathcal{J} - \mathcal{J}_{-i}$, where $\mathcal{J}_{-i}$ is the cost of the same solution with ULD i removed. The best $\mathbf{q}$ ULDs are removed.
- "least removed" removal: we keep track of the number of times each ULD has been removed from the current solution, and we remove from $\mathcal{S}$ the $\mathbf{q}$ ULDs that were removed less frequently.
- smallest route removal: the route with the least number of ULDs is removed from $\mathcal{S}$. In this case, the number of removed ULDs might differ from $\mathbf{q}$. This removal is added to potentially steer the solution towards a lower number of used trucks, pending the feasible re-insertion of the ULDs in one of the remaining routes.

Next, insertion operators are summoned to reconstruct a feasible solution. We consider the following:

- greedy insertion heuristic: the initial solution algorithm, starting from a non-empty solution.
- tabu greedy insertion heuristic: this heuristic is based on the same concept as the greedy insertion heuristic, but keeps track of the last known route an unassigned ULD was part of, and prevents the ULD to be re-assigned to it.
- 2-regret insertion heuristic: this heuristic leverages the possible myopic effect of the greedy insertion, that looks for a short-term reward without necessarily considering long-term effects. Let us define $\Delta c_{i,1}$ the increase in the cost function when inserting ULD i in the best available route (in the best insertion point) and $\Delta c_{i,2}$ the increase in the cost function when inserting ULD i in the second best available route (in the best insertion point). If a ULD can only be feasibly inserted in one route, then we set $\Delta c_{i,2} = \infty$. We select the ULD that would cause the greatest increase in the objective if the first opportunity (best insertion route) is missed.
- route creation: a greedy insertion heuristic is applied to create a new route using as many unassigned ULDs as possible. Similarly to the smallest route removal, this insertion move is useful to modify the number of trucks used. In addition, since we do not specify any fixed cost associated to trucks, increasing the number of trucks is beneficial if this helps reducing the overall traveling time.

Table 3
Instance characteristics in terms of FFs, GHs, and ULDs.

Instance type	# FF	# GH	# ULD	# docks
Small	U(2,3)	U(2,3)	U(8,20)	1
Medium	U(3,4)	U(3,4)	U(20,40)	1
Large	U(4,5)	U(4,5)	U(40,70)	1
Very large	7	U(4,5)	U(65,90)	2

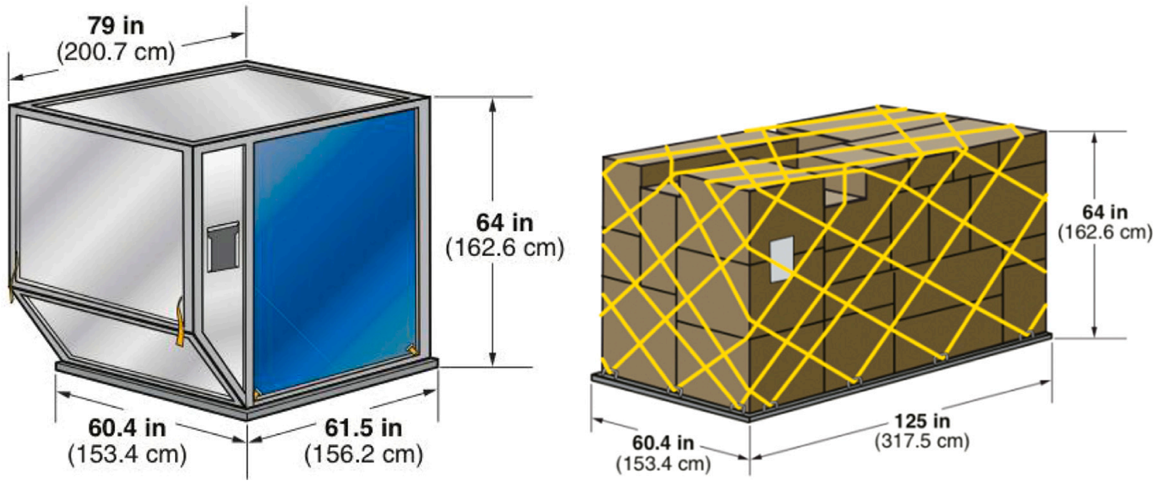

Fig. 4. ULDs considered in the instances (left: LD-3, right: pallet). From www.searates.com.

After each insertion move, we run the dock capacity resolver routine to detect and possibly solve violations. Given the revised truck schedules forming solution $\mathcal{S}'$, we compute the new $\mathcal{J}(\mathcal{S}')$ according to Eq. (62).

With regard to the choice of a specific removal and insertion operator, initially all operators have the same probability. If the removal–insertion resulted in a new accepted solution, this probability is increased by a certain factor based on the goodness of the solution. After a set number of iterations, named segment, the probabilities are reevaluated based on the performance within the previous segment.

Finally, the new solution $\mathcal{S}'$ is evaluated according to the SA acceptance criterion (Boltzmann distribution). In particular, the solution is accepted according to the probability $P(T) = \min\left\{ e^{\frac{(\mathcal{J}(\mathcal{S})-\mathcal{J}(\mathcal{S}'))}{T}}, 1 \right\}$. The parameter T, called cooling temperature, is set initially in such a way that a solution $w\%$ worse than $\mathcal{S}_0$ is accepted with a 0.5 probability. In the next iterations it is reduced via a cooling factor $T = c \cdot T$. The algorithm is stopped either when the maximum number of iterations is reached, or when the temperature reaches zero.

5. Numerical experiments

5.1. General description of instances

We generate synthetic yet realistic instances based on academic literature, in particular on the work by Ankersmit et al. (2014), and on discussions with the cargo department of a partner airline and other stakeholders. We focus on Amsterdam Airport Schiphol (AMS), one of the most important European cargo gateways, and selected 7 FFs located in the surroundings of the airport and 5 major GHs. We divided instances in 4 categories (*small*, *medium*, *large*, and *very large*) according to the number of warehouses involved, ULDs, and docks. See Table 3 for an overview.

Two ULD types can be generated with equal probability: LD-3 containers and pallets. The two ULD types differ in length, weight, and processing time; see Fig. 4 for details. Note that pallets cannot be rotated to only occupy 1.54 m, since trailers are generally shorter than 3.18 m in width. The weights are randomly chosen from normal distributions, $\mathcal{N} \sim (1000, 2000)$ kg and $\mathcal{N} \sim (2800, 4000)$ kg respectively for LD-3s and pallets. Processing times are generated using triangular distributions, $\mathcal{T} \sim (2, 4, 7)$ minutes and $\mathcal{T} \sim (4, 7, 10)$ minutes respectively for LD-3s and pallets. For simplicity, both measures are rounded to the next integer. We consider a planning horizon for daily operations of 480 min. With regard to time windows, we generate for pickup nodes only the earliest moment of visit and for delivery nodes only the latest one, since these two are the most relevant in this setting as explained in Section 3.1.

We consider a homogeneous fleet where each trailer has a maximum weight capacity of $Q^W = 10,000$ kg and a maximum length of $Q^L = 13.6$ m. For each instance, we consider a fleet size equal to $|\mathcal{K}| = \max\left[\frac{\sum_{i\in\mathcal{V}} W_i}{Q^W}, \frac{\sum_{i\in\mathcal{V}} L_i}{Q^L}\right] + 1$. To compute traveling times and

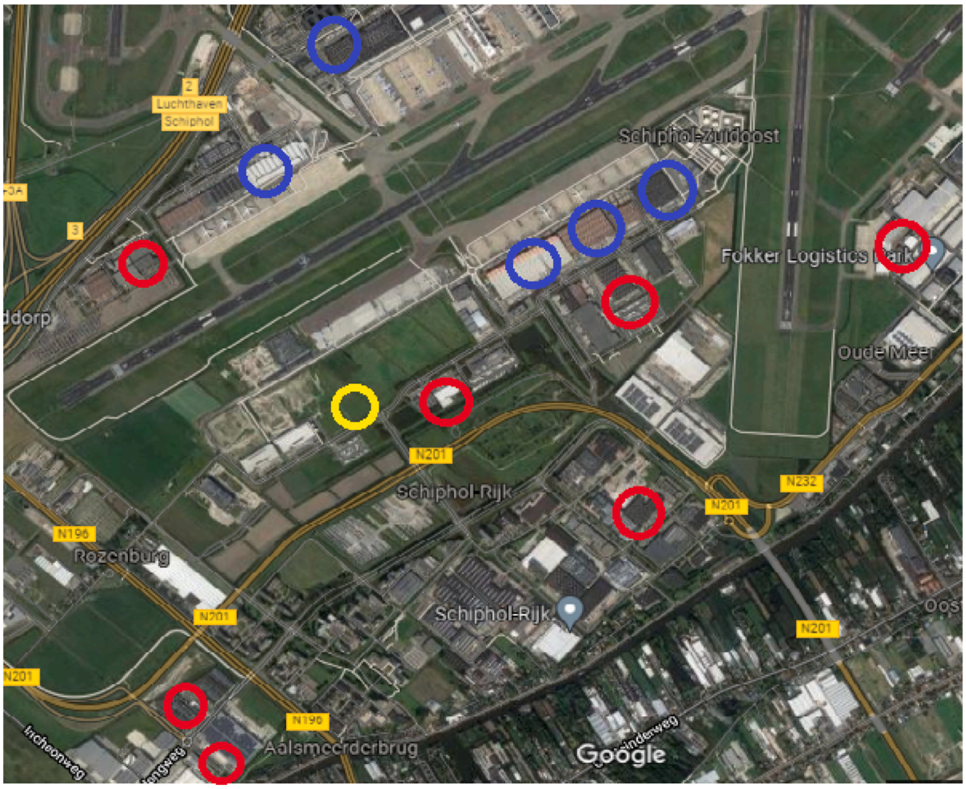

Fig. 5. Location of the FFs (red), GHs (blue), and central depot (yellow) around AMS. From https://www.google.com/maps. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

to guarantee the triangular inequality, we determine the Euclidean distance between all warehouses, double the value and divide by an average speed of 35 km/h. We also position the central depot in a currently empty area just outside AMS and that is central with respect to all the warehouses involved. It should be noted that determining an optimal location for such a depot is outside the scope of the paper. We show the location of the FFs, the GHs, and the central depot in Fig. 5. We set a maximum of two docks per GH. This choice is reasonable if we consider simultaneously (i) the size of our consortium when compared to all the possible FFs serving a GH and (ii) that GHs have, on average, no more than 20 docks for export operations (Ankersmit et al., 2014).

We define three classes for instances generation: 20%, 50%, 80%. Class 20% entails that 20% of randomly selected ULDs in an instance will be characterized by a tighter time window than $[0, 480]$ min. Since each ULD is mapped both with a pickup and a delivery node, we impose tighter time windows on both 20% of pickup and delivery nodes. For pickup nodes we randomly select the earliest pickup time between 50 and 150 min, whereas for delivery nodes a due date between 200 and 350 min. When both time windows of the same ULD are tightened, we ensure feasibility by extending the second time window by 30 min. For each combination of instance size and time window percentage, we generated 5 instances for a total of 45 instances. We use the following notation: `ID_FF_GH_ULD_PERC-TW_ND`, where `ID` is the identifier of the instance within the same block (instance size, time window percentage), `FF` and `GH` are the number of FFs and GHs, respectively, `ULD` is the number of ULDs, `PERC-TW` is the time window percentage, and `ND` is the number of docks.

5.2. Metaheuristic parametrization and technical settings

The metaheuristic is coded in Python and run on an Intel(R)Core(TM)2 DUO machine with 2.93 GHz and 8.00 GB RAM memory. The two MILP models are integrated in CPLEX using the Concert Technology in C++ and each instance was run for a maximum of 1 h (small instances), 6 h (medium instances), and 9 h (large and very large instances) on a high performing machine, with a 24 Intel Xeon 2.5 GHz cores and 128 GB of internal memory, using twenty-four threads in default conditions.

For the metaheuristic, the maximum number of iterations is set to 7500 and we define a segment as a sequence of 50 iterations. For the first segment, all operators have the same probability to be chosen (20% for removal, 25% for insertion operators). Finally, we defined $\alpha = \beta = 100$ to penalize dock capacity violations and non-delivered ULDs in our definition of the cost function in Eq. (62). With regard to the number $\mathbf{q}$ of ULDs to remove, we set $\mathbf{q} = \min\left[\left\lceil \frac{\sigma}{3} \right\rceil, \left\lceil \frac{|C_{\Uparrow}|}{5} \right\rceil\right]$. Other parameters are in line with the setting in Ropke and Pisinger (2006).

Table 4
Results for small instances.

Instance	Percentage time windows: 20%											
	M_1				M_2				ALNS			
	Best inc.	LB	Gap [%]	Time [s]	Best inc.	LB	Gap [%]	Time [s]	min.	avg.	max.	Time [s]
1_2_2_8_0.2_1	132.9	132.9	0.0	6	132.9	132.9	0.0	68	132.9	132.9	132.9	87
2_3_2_14_0.2_1	222.1	222.1	0.0	578	222.1	222.1	0.0	107	222.1	222.1	222.1	143
3_3_2_16_0.2_1	240.3	240.3	0.0	2712	240.3	240.3	0.0	198	240.3	240.3	240.3	152
4_2_3_15_0.2_1	203.0	203.0	0.0	2915	203.0	203.0	0.0	117	203.0	203.0	203.0	172
5_2_3_14_0.2_1	209.6	209.6	0.0	2119	209.6	209.6	0.0	114	209.6	209.6	209.6	193
	Percentage time windows: 50%											
1_2_2_9_0.5_1	140.4	140.4	0.0	18	140.4	140.4	0.0	72	140.4	140.4	140.4	92
2_3_2_16_0.5_1	239.4	239.4	0.0	1286	239.4	239.4	0.0	114	239.4	239.4	239.4	163
3_3_2_15_0.5_1	245.9	245.9	0.0	3073	245.9	245.9	0.0	115	245.9	245.9	245.9	176
4_2_3_14_0.5_1	200.3	200.3	0.0	1133	200.3	200.3	0.0	104	200.3	200.3	200.3	157
5_2_3_13_0.5_1	193.6	193.6	0.0	377	193.6	193.6	0.0	240	193.6	194.5	196.2	163
	Percentage time windows: 80%											
1_2_2_11_0.8_1	168.4	168.4	0.0	21	168.4	168.4	0.0	78	168.4	168.4	168.4	90
2_3_2_15_0.8_1	225.9	225.9	0.0	2322	225.9	225.9	0.0	189	225.9	227.3	229.1	143
3_3_2_14_0.8_1	217.4	217.4	0.0	551	217.4	217.4	0.0	111	217.4	217.4	217.4	197
4_2_3_16_0.8_1	224.7	224.7	0.0	2830	224.7	224.7	0.0	111	224.7	225.8	228.3	175
5_2_3_14_0.8_1	206.6	206.6	0.0	2520	206.6	206.6	0.0	107	206.6	206.6	206.6	139

With regard to the models, they rely quite heavily on big-M constants, both for time precedence constraints and to define arrivals/departures of trucks to/from GHs. Big-M constants are known to induce high optimality gaps and, hence, increase computational times. To limit this issue, we adopted well-known (see Cordeau (2006)) techniques to minimize the needed big-M for time precedence constraints. On a similar note, we used upper and lower bounds on arrival/departure times at GHs to keep big-M constants as small as possible in the associated constraints.

5.3. Comparison of M_1, M_2, and the ALNS

We compare the performances with a set of 60 instances, setting the number of docks to 2 for very large instances and to 1 otherwise. In Tables 4, 5, 6, and 7 we report the results. For small instances, our goal was to ensure consistency between the three methods. For the other instances, the primary goal was twofold. First, to compare the best results of the three methods in case an optimal solution was not obtained. Second, to assess the maximum problem size that our MILP formulations could tackle within a reasonable computational time. Results are reported in terms of best (min.), average, and worst (max.) objective. For the MILP formulations, **Best Inc.** is the best feasible solution in the branch-and-bound tree, whereas LB is the best lower bound. Finally, we define **Gap** as **(Best inc.-LB)/(Best inc.)$*$ 100**. For the ALSN, each instance was solved 10 times with different random seeds and **Time** defines the rounded average time across the 10 runs until one of the stopping criteria is met.

For small instances, all three methods converged to the optimal solution. Except for the instances starting with `1_` (that are the smallest per block), M_2 proved to be faster than M_1 in obtaining the optimal solution. We believe the strong reduction of routing variables to be the key in confining the solution space and making M_2 converge quicker. Our ALNS converged to the optimal solution as well, but for small instances the exact formulation M_2 was faster. We believe that the additional computational overhead of the metaheuristic makes it not as efficient as a state-of-the-art branch-and-bound solver for such instances.

Medium instances provided interesting insights into the comparison between M_1 and M_2. In fact, M_2 was the only formulation to solve three instances to optimality. For the same instances, M_1 computed a best incumbent that is very close to the optimal solution, but with a larger optimality gap. This trend is identifiable for the whole set of medium instances. Even when M_1 provided a final best incumbent, M_2 provided a tighter gap. Differently from the small instances, the ALNS displays differences in terms of final best solution for the more demanding medium instances (see Table 5), although the dispersion around the mean value is quite limited. In addition, results are comparable with the best incumbent obtained from one of the two exact formulations, but with a much shorter computational time.

For large instances, M_1 outperforms M_2 when it comes to finding an integer solution. In particular, M_1 manages to find a solution for 13 out of 15 large instances, while M_2 only finds a best incumbent in 4 cases. On the other hand, and consistently with the trend described for smaller instances, M_2 has a tighter linear relaxation and provides a larger lower bound for almost all instances. Consistently with medium instances and with the results of the ALNS algorithm, we believe the integer solutions to be reasonably close to the optimal value. For large instances, our ALNS algorithm performed reasonably well, finding for two instances a slightly better solution that the exact formulations. On the downside, the algorithm is less stable and a greater variability in the final solution is achieved, as evidenced by Table 6. For the two instances where neither M_1 nor M_2 found a feasible solution, the ALSN did not manage to find a feasible solution either. As we will be elaborating more in Section 5.5, we believe that the two instances are infeasible due to the large number of ULDs characterized by a tighter delivery window and the presence of only 1 dock per GH to accommodate such demand.

Table 5
Results for medium instances.

Instance	Percentage time windows: 20%											
	M_1				M_2				ALNS			
	Best inc.	LB	Gap [%]	Time [s]	Best inc.	LB	Gap [%]	Time [s]	min.	avg.	max.	Time [s]
1_3_3_23_0.2_1	358.3	307.7	14.1	21,600	355.3	355.3	0.0	5077	355.3	357.8	368.3	543
2_4_3_30_0.2_1	481.3	387.1	19.6	21,600	474.8	446.3	6.0	21,600	476.2	480.2	501.2	668
3_4_3_27_0.2_1	429.9	365.7	17.0	21,600	419.0	385.2	8.2	21,600	419.0	421.0	430.1	613
4_4_4_38_0.2_1	608.1	488.5	19.7	21,600	598.3	504.9	15.6	21,600	599.1	608.3	621.3	1013
5_4_4_39_0.2_1	675.9	505.6	25.2	21,600	664.9	538.9	19.0	21,600	665.0	670.8	701.4	986
	Percentage time windows: 50%											
1_3_3_21_0.5_1	319.0	261.1	18.1	21,600	311.8	311.8	0.0	1505	311.8	314.3	320.0	613
2_4_3_29_0.5_1	474.7	377.7	20.4	21,600	469.2	420.7	10.3	21,600	470.1	480.2	497.3	798
3_4_3_29_0.5_1	469.2	376.0	19.9	21,600	465.2	418.1	6.1	21,600	465.2	480.9	492.7	853
4_4_4_38_0.5_1	650.6	516.7	20.6	21,600	659.4	561.8	14.8	21,600	661.3	680.1	695.3	1143
5_4_4_37_0.5_1	640.3	489.2	23.6	21,600	630.2	535.8	15.0	21,600	631.3	640.1	693.2	1021
	Percentage time windows: 80%											
1_3_3_23_0.8_1	369.9	301.4	18.5	21,600	360.4	360.4	0.0	18,917	362.3	367.1	382.9	601
2_4_3_32_0.8_1	528.1	425.4	19.4	21,600	511.1	471.1	7.8	21,600	516.3	533.2	551.8	832
3_4_3_29_0.8_1	480.8	390.5	18.8	21,600	473.3	435.4	8.0	21,600	476.1	491.5	518.7	908
4_4_4_37_0.8_1	628.9	484.0	23.0	21,600	630.1	526.2	16.5	21,600	631.3	642.3	661.4	1301
5_4_4_39_0.8_1	652.7	481.1	26.3	21,600	645.7	525.2	18.7	21,600	650.3	672.4	721.6	1209

Table 6
Results for large instances.

Instance	Percentage time windows: 20%											
	M_1				M_2				ALNS			
	Best inc.	LB	Gap [%]	Time [s]	Best inc.	LB	Gap [%]	Time [s]	min.	avg.	max.	Time [s]
1_5_4_48_0.2_1	787.0	595.0	24.4	32,400	774.4	610.1	21.2	32,400	775.0	791.3	823.7	1923
2_5_4_50_0.2_1	800.2	601.4	24.8	32,400	795.2	627.3	21.1	32,400	798.4	807.3	841.3	2016
3_5_5_64_0.2_1	1023.2	775.8	24.2	32,400	–	763.8	–	32,400	1012.3	1041.8	1104.0	3804
4_5_5_66_0.2_1	1056.8	773.4	26.8	32,400	–	781.5	–	32,400	1062.2	1081.5	1137.2	2592
5_5_5_63_0.2_1	1013.3	790.8	22.0	32,400	–	785.5	–	32,400	1041.4	1071.7	1185.2	2326
	Percentage time windows: 50%											
1_5_4_49_0.5_1	822.3	614.7	25.2	32,400	818.4	637.7	22.1	32,400	821.3	837.1	873.4	1783
2_5_4_50_0.5_1	812.7	612.4	24.6	32,400	–	627.6	–	32,400	812.9	845.3	889.6	1836
3_5_5_61_0.5_1	1010.7	760.6	24.7	32,400	–	768.8	–	32,400	1018.3	1051.3	1105.0	2502
4_5_5_67_0.5_1	1031.0	785.7	23.8	32,400	–	799.5	–	32,400	1014.8	1065.1	1178.2	4011
5_5_5_60_0.5_1	995.8	734.5	26.2	32,400	–	722.2	–	32,400	1003.1	1050.2	1131.9	2832
	Percentage time windows: 80%											
1_5_4_50_0.8_1	801.9	615.7	23.2	32,400	–	631.9	–	32,400	803.3	843.2	895.4	1673
2_5_4_51_0.8_1	845.4	599.5	29.1	32,400	804.6	616.3	23.4	32,400	813.1	871.4	912.2	1803
3_5_5_67_0.8_1	–	783.1	–	32,400	–	809.2	–	32,400	N/A	N/A	N/A	3127
4_5_5_66_0.8_1	–	760.6	–	32,400	–	777.5	–	32,400	N/A	N/A	N/A	3018
5_5_5_59_0.8_1	973.4	725.0	25.5	32,400	–	712.6	–	32,400	981.2	1013.2	1123.5	2912

Finally, for very large instances models M_1 and M_2 struggled to find even a feasible solution. Only M_1 did find a best incumbent for three instances (the smallest instance per time-window block). However, M_2 proved again to provide a slightly tighter linear relaxation, as testified by the generally higher values of **LB**. In addition, the routing-based approach of M_1 scaled badly, as expected, when increasing the number of ULDs to values greater than 70. Out of the 15 very large instances, in ten cases M_1 ran out of memory (rows with an asterisk in the **Time** column). Our ALSN managed to find a feasible solution for all instances instead. In particular, for the three cases where we had a comparison with M_1, in two of them it managed to find a lower overall transportation time. The variability of the solution quality across different runs, a known issue of this type of metaheuristic, increased as the size of the instances increased.

5.4. Comparison between the non-cooperative and the cooperative framework

In this experiment, we compare the cooperative framework against the non-cooperative one for medium, large, and very large instances. In order to compute the non-cooperative solution, for each instance with $|\mathcal{F}|$ FFs, we disaggregate the input data creating $|\mathcal{F}|$ sub-instances, where in every sub-instance a single FF i optimizes the delivery of its own subset of ULDs. Once all sub-instances are solved, we aggregate the solutions using a First-Come-First-Served (FCFS) approach. We use the FCFS approach to replicate what would happen in a non-cooperative framework, i.e., that trucks arriving at a GH must queue following the order they arrived and

Table 7
Results for very large instances.

Instance	Percentage time windows: 20%											
	M_1				M_2				ALNS			
	Best inc.	LB	Gap [%]	Time [s]	Best inc.	LB	Gap [%]	Time [s]	min.	avg.	max.	Time [s]
1_7_4_67_0.2_2	1125.9	801.3	28.8	32,400	–	806.9	–	32,400	1168.3	1195.3	1245.0	3542
2_7_4_72_0.2_2	–	843.3	–	6580*	–	870.4	–	32,400	1252.9	1307.1	1413.2	3853
3_7_5_90_0.2_2	–	1058.9	–	5660*	–	1088.9	–	32,400	1492.7	1574.1	1702.3	4823
4_7_5_86_0.2_2	–	1005.0	–	4548*	–	999.8	–	32,400	1370.2	1480.9	1572.3	4927
5_7_5_87_0.2_2	–	1028.0	–	4747*	–	1057.1	–	32,400	1460.2	1548.8	1613.9	5013
	Percentage time windows: 50%											
1_7_4_70_0.5_2	–	864.5	–	32,400	–	877.8	–	32,400	1262.1	1301.3	1392.4	3427
2_7_4_67_0.5_2	1225.4	857.2	30.0	32,400	–	841.4	–	32,400	1193.6	1220.4	1315.4	3294
3_7_5_89_0.5_2	–	1054.0	–	3584*	–	1075.6	–	32,400	1530.1	1610.4	1725.6	5442
4_7_5_87_0.5_2	–	1040.0	–	3488*	–	1062.5	–	32,400	1437.8	1512.4	1616.2	4943
5_7_5_86_0.5_2	–	1037.6	–	4557*	–	1046.6	–	32,400	1329.1	1486.1	1567.9	5126
	Percentage time windows: 80%											
1_7_4_71_0.8_2	–	851.2	–	32,400	–	870.8	–	32,400	1160.1	1238.4	1315.8	3927
2_7_4_71_0.8_2	1271.1	856.9	32.6	32,400	–	883.3	–	32,400	1219.0	1294.7	1384.1	4193
3_7_5_80_0.8_2	–	960.0	–	7116*	–	947.2	–	32,400	1342.0	1377.3	1472.4	4572
4_7_5_87_0.8_2	–	1027.6	–	10,927*	–	1044.5	–	32,400	1392.3	1516.8	1621.3	5236
5_7_5_85_0.8_2	–	1021.0	–	4883*	–	1040.0	–	32,400	1327.3	1480.4	1592.5	4912

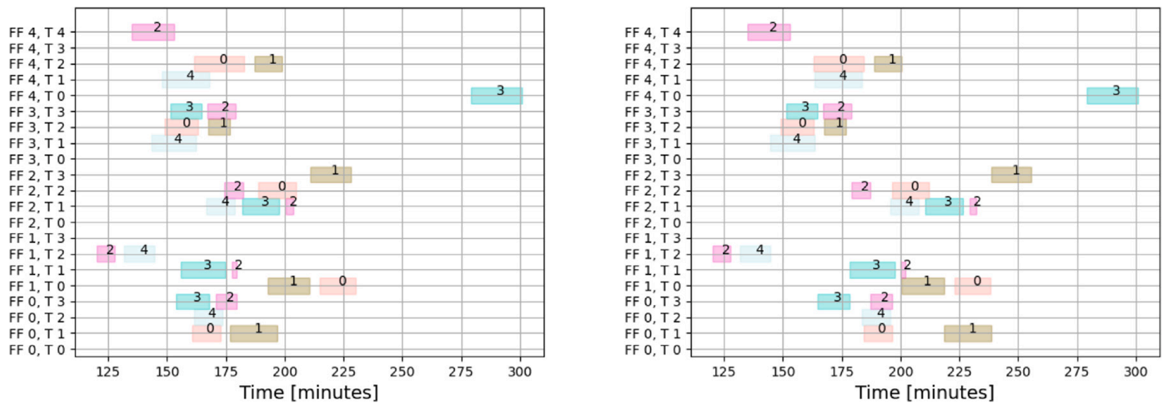

Fig. 6. Example of non-cooperative solution before (left) and after (right) FCFS schedule revision for instance 3_5_5_67_0.8_1. Numbers in the bars define the specific GH.

wait until a dock is made available. Note that this aggregation may lead to worse or infeasible solutions in case trucks are delayed. Fig. 6 shows an example of a non-cooperative scheduling solution for the GH side before (left) and after (right) FCFS aggregation for instance 3_5_5_67_0.8_1.

In Table 8, we show the results for the two settings in terms of overall traveling time OTT, number of trucks used $|\mathcal{K}|$, and overall delay experienced by trucks while waiting for a dock OWT. We report the best solution found by the three solving methods and for the non-cooperative solution, we point out whether the aggregation resulted in a loss of feasibility.

The analysis of Table 8 can be carried out using different perspectives. First, in terms of OWT values, within the cooperative framework only in 5 cases trucks experience delays; whereas for the non-cooperative framework they do in all cases but one. The delays in the latter case reached even 3 h for the most demanding instances. In addition, for some instances, the non-cooperative frameworks displays a traveling time between warehouses that is lower than the associated traveling time of the cooperative framework due to lack of routing options on the FF side, but the much larger OWT makes the final objective worse.

Secondly, as it concerns the feasibility of the aggregated non-cooperative solutions, most of the medium instances turned out to provide a feasible solution except the last two. This was expected since with 80% of ULDs characterized by a tighter time window, the chances of missing a delivery due to the FCFS approach significantly increase. This concept appears even more explicitly for large instances. While instances with 20% of tighter time windows are still all feasible, two instances for the 50% and four for the 80% case are infeasible. The latter case is also where accumulated delays are the largest. For very large instances, only one instance belonging to the 80% case is infeasible in the non-cooperative scenario. This is reasonable due to the availability of the second dock: queues and, hence, delays will be generated only if 3 or more FFs are visiting the same GH simultaneously. This is also testified by the relatively smaller average OWT of very large instances when compared to large instances.

Table 8
Comparison between cooperative and non-cooperative solution for medium, large and very large instances.

Instance	Cooperative			Non-cooperative				Comparison	
	OTT [min.]	$\|\mathcal{K}\|$	OWT [min.]	OTT [min.]	$\|\mathcal{K}\|$	OWT [min.]	Feasible [Y/N]	ΔOTT [%]	$\Delta\|\mathcal{K}\|$ [%]
1_3_3_23_0.2_1	355.3	4	0.0	381.1	5	19.2	Y	−6.8	−20
2_4_3_30_0.2_1	474.8	6	0.0	522.4	8	16.1	Y	−9.1	−25
3_4_3_27_0.2_1	419.0	5	0.0	481.0	8	24.5	Y	−12.9	−38
4_4_4_38_0.2_1	614.6	8	0.0	620.6	8	22.6	Y	−1.0	0
5_4_4_39_0.2_1	664.9	9	0.0	682.4	10	12.9	Y	−2.6	−10
1_3_3_21_0.5_1	311.8	4	0.0	365.0	6	30.1	Y	−14.6	−33
2_4_3_29_0.5_1	474.7	6	0.0	488.8	7	5.2	Y	−2.9	−14
3_4_3_29_0.5_1	465.2	6	0.0	513.9	7	32.3	Y	−9.5	−14
4_4_4_38_0.5_1	650.6	8	0.0	683.8	9	17.4	Y	−4.9	−11
5_4_4_37_0.5_1	640.3	8	0.0	655.7	9	21.9	Y	−2.4	−11
1_3_3_23_0.8_1	360.4	5	0.0	371.7	6	0.0	Y	−3.0	−17
2_4_3_32_0.8_1	520.2	6	0.0	636.5	8	96.0	Y	−18.3	−25
3_4_3_29_0.8_1	480.8	6	0.0	504.8	7	20.1	Y	−4.8	−14
4_4_4_37_0.8_1	628.9	8	0.0	688.9	8	70.7	N	−9.6	0
5_4_4_39_0.8_1	652.7	9	1.3	771.1	9	129.5	N	−15.4	0
1_5_4_48_0.2_1	774.4	10	0.0	792.6	11	10.1	Y	−2.3	−9
2_5_4_50_0.2_1	795.2	10	0.0	810.5	11	31.1	Y	−1.9	−9
3_5_5_64_0.2_1	1012.3	13	0.0	1085.3	15	35.6	Y	−6.5	−13
4_5_5_66_0.2_1	1056.8	13	0.0	1108.8	15	78.9	Y	−4.7	−13
5_5_5_63_0.2_1	1013.3	14	25.6	1074.2	15	33.9	Y	−5.7	−7
1_5_4_49_0.5_1	818.4	10	0.0	913.6	10	136.6	N	−10.4	0
2_5_4_50_0.5_1	812.7	10	0.0	861.5	12	67.8	Y	−5.7	−17
3_5_5_61_0.5_1	1010.7	13	0.0	1011.1	13	27.5	Y	−0.1	0
4_5_5_67_0.5_1	1014.8	14	0.0	1108.2	15	47.5	N	−8.4	−7
5_5_5_60_0.5_1	995.8	12	0.0	999.8	13	31.0	Y	−0.4	−8
1_5_4_50_0.8_1	801.9	10	0.0	896.7	12	95.4	N	−10.6	−17
2_5_4_51_0.8_1	804.6	11	0.0	810.5	11	31.1	Y	−0.7	0
3_5_5_67_0.8_1	–	–	–	1271.7	16	193.3	N	–	–
4_5_5_66_0.8_1	–	–	–	1094.5	15	59.6	N	–	–
5_5_5_59_0.8_1	973.4	12	0.0	1075.3	14	136.0	N	−9.5	−14
1_7_4_67_0.2_2	1125.9	13	0.0	1196.2	13	12.2	Y	−5.9	0
2_7_4_72_0.2_2	1259.9	15	0.0	1347.2	16	22.4	Y	−7.0	−6
3_7_5_90_0.2_2	1492.7	18	19.4	1672.5	20	39.5	Y	−10.8	−10
4_7_5_86_0.2_2	1370.2	17	0.0	1421.2	18	31.2	Y	−3.6	−6
5_7_5_87_0.2_2	1460.2	17	13.1	1573.6	18	40.4	Y	−7.2	−6
1_7_4_70_0.5_2	1262.1	14	0.0	1280.0	14	23.8	Y	−1.4	0
2_7_4_67_0.5_2	1193.6	13	0.0	1217.3	14	28.5	Y	−5.7	−7
3_7_5_89_0.5_2	1530.1	17	21.4	1603.2	19	47.1	Y	−1.9	−10
4_7_5_87_0.5_2	1437.8	17	0.0	1562.2	18	48.9	Y	−8.0	−6
5_7_5_86_0.5_2	1329.1	16	0.0	1572.4	17	81.0	Y	−15.4	−6
1_7_4_71_0.8_2	1160.1	14	0.0	1332.9	15	15.4	Y	−13.0	−7
2_7_4_71_0.8_2	1219.0	14	0.0	1284.1	15	31.1	Y	−5.0	−7
3_7_5_80_0.8_2	1342.0	15	0.0	1485.5	16	63.3	Y	−9.6	−6
4_7_5_87_0.8_2	1392.3	17	0.0	1532.2	19	97.6	N	−9.1	−11
5_7_5_85_0.8_2	1327.3	17	0.0	1459.9	18	76.0	Y	−9.0	−6

Finally, with respect to the overall gain due to the cooperative framework, we can focus on ΔOTT and $\Delta|\mathcal{K}|$. There were a few cases where the cooperation did not provide any substantial advantage, neither in terms of overall traveling time nor in terms of fleet utilization. This could be due to the sub-optimality of the solution of our framework, especially for large and very large instances. For the other cases, reductions in OTT range from 2 to 19%. As expected, the reduction in terms of fleet utilization is much larger, due to the possibility to increase load factors by mixing ULDs from different FFs headed to the same GH. Interestingly, the cases where the cooperation does not improve OTT are also characterized by a $\Delta|\mathcal{K}| = 0$.

5.5. Analysis of Δ_{GH}

In this experiment, we aim to analyze the impact of different number of docks at the GHs reserved to the consortium (Δ_{GH}). For large instances characterized by a time window percentage of 80%, we set Δ_{GH} to 1 or 2. We decided to focus solely on these instances since only in 3 out of 5 cases a feasible solution could be found for $\Delta_{GH} = 1$. We did not perform a similar analysis for very large instances, since our intuition was that most of them would have been infeasible when $\Delta_{GH} = 1$.

In Table 9 we report the comparison of the best objective when $\Delta_{GH} = 1$ and $\Delta_{GH} = 2$. For both cases, we report the best solution (expressed similarly to Table 8 as OTT) between M_1, M_2, and the ALSN. When $\Delta_{GH} = 2$, for the two exact formulations we warm-started the branch-and-bound solver with the best incumbent (if available) obtained when $\Delta_{GH} = 1$. As a matter of fact, a feasible solution for $\Delta_{GH} = 1$ is still feasible when $\Delta_{GH} = 2$ with the second dock trivially not used at all.

Table 9
Comparison between $\Delta_{GH} = 1$ and $\Delta_{GH} = 2$ for large instances with a time window percentage of 80%.

Instance	$\Delta_{GH} = 1$		$\Delta_{GH} = 2$		Comparison	
	OTT [min.]	$\lvert\mathcal{K}\rvert$	OTT [min.]	$\lvert\mathcal{K}\rvert$	ΔOTT [%]	$\Delta\lvert\mathcal{K}\rvert$ [%]
1_5_4_50_0.8	801.9	10	798.3	10	−0.5	0
2_5_4_51_0.8	804.6	11	804.6	11	0.0	0
3_5_5_67_0.8	–	–	1145.9	14	–	–
4_5_5_66_0.8	–	–	1152.6	14	–	–
5_5_5_59_0.8	973.4	12	946.9	12	−2.7	0

The analysis of Table 9 leads to the following conclusions. First, for the instances where a feasible solution was found when $\Delta_{GH} = 1$, there is not a strong incentive in doubling the number of available docks. Keeping in mind that no optimal solution was found, for the 3 cases that were already solved, the maximum decrease in the OTT time is 2.3%. For one instance, we could not even improve the solution obtained with $\Delta_{GH} = 1$ at all. On the contrary, for instances `3_5_5_67_0.8_1` and `4_5_5_66_0.8_1`, where no method managed to find a feasible solution when $\Delta_{GH} = 1$, a feasible solution was found. While the branch-and-bound tree was not fully explored when $\Delta_{GH} = 1$, and hence our claim cannot be fully corroborated, we believe these two instances to be infeasible with only one dock per GH. Not surprisingly, they are the two instances with the highest number of ULDs.

Finally, for the case $\Delta_{GH} = 2$, we investigate how often the docks dedicated to the consortium are occupied simultaneously by two trucks. Given the aforementioned considerations, we focused on instances `3_5_5_67_0.8_2` and `4_5_5_66_0.8_2`. Both docks are simultaneously occupied only for a very limited time, most likely to perform delivery of ULDs with tighter time windows that could not be performed by the same truck. Simultaneous use of both docks at the same GH occurs in less than 10% of the overall dock occupation time. This provides ample opportunities to the central planner to make the second dock available to other trucks not belonging to the consortium, for example using a time-slot approach similar to the one implemented by Bruxelles airport (Nallian, 2021). All in all, it seems like the addition of the second dock is only exploited to resolve those conflicts that would be infeasible with a single dock scenario. With more FFs or more ULDs, the presence of the second dock becomes necessary, but the advantage of our approach when compared to a slot booking-based approach is that available slots are easily identifiable and auctionable to FFs outside the consortium.

6. Conclusions

In this paper, we have proposed a novel framework for the cooperation between freight forwarders who need to send air cargo in the form of ULDs to ground handlers. Freight forwarders may use a shared fleet of trucks that first pick up their ULDs and then deliver them to the associated ground handlers. The cooperation can potentially limit the congestion at the ground handlers docking stations since these have typically small handling capacity due to limited space at the air terminals. In our proposed approach, we assume that every ground handler allocates a subset of its export docks only for the members of such a consortium.

The problem can be modeled as a hybrid version of the capacitated vehicle routing problem with pickup and delivery and parallel machine scheduling problem. The former is related to the routing of the trucks, the latter to their sequence and scheduling at the ground handlers' docks. We developed two mathematical models for the problem, one fully based on the routing structure of the problem and the other making use of bin packing variables. When tested on CPLEX, the second model showed a better average performance (MILP gap), however it struggled to find integer solutions for larger instance sizes. To overcome the complexity of the problem, we have also developed an Adaptive Large Neighborhood Search metaheuristic that could provide overall comparable or better solutions in less computational time.

With a set of numerical experiments we have also quantified the benefits of this cooperation against the individualistic case. In the latter case, freight forwarders have no notion of the scheduling of the other parties. This individualistic planning may contribute to congestion, but also to the risk of exceeding due dates of the ULDs. This turned out to be particularly true for large instances, where most non-collaborative solutions turned out to be infeasible, especially when most ULDs were characterized by tighter time windows. This aspect is particularly crucial in today's cargo operations that rely more and more on a just-in-time approach to limit stocking costs. Hence, we believe our collaborative framework can provide an effective modus operandi to ensure seamless export operations.

For most instances, we relied on a single dock per ground handler. This proved to be sufficient to satisfy demand and time windows for most large instances. For some instances, we only managed to compute a feasible solution when increasing the number of docks available to two. Note that, due to the centralized planning approach, reserving two docks to trucks belonging to the consortium does not mean those docks will only be used exclusively during the planning horizon. In fact, ample time slots (according to the specific due dates and requirements of the ULDs) can be identified to be auctioned to other freight forwarders not belonging to the consortium, to further improve the cargo throughput. In this sense, we believe our approach not only optimizes the schedule of the neutral fleet and improves on-time performances, but also wisely uses the available resources and identifies opportunities to re-offer those resources to the other stakeholders involved in the supply chain.

The proposed setting has not received much attention in the literature and this paper is the first attempt to highlight the operational challenges of this complex transport system. Future research may further extend the problem at hand with interesting modeling extensions or explore tailored solution methods. For example, it would be interesting to use a combination of column-and row-generation techniques. With the former, our arc-based formulation would be translated into a path-based formulation with the overarching goal of improving lower bounds. The latter is also needed to identify scheduling conflicts and avoid them adding constraints on the fly. In addition, while we used deterministic input parameters, most of them are stochastic in nature, and hence modeling this aspect might be interesting, either via model extensions or within a simulation framework.

From a practical perspective, it is crucial to define the conditions for which FFs will participate in such a consortium. A focus on how transportation costs will be eventually split among the different freight forwarders is necessary to ensure the economic, together with the operational, viability of such an approach. In case of revenue-sharing (on top of cost-sharing) agreements, profit-sharing models can be devised to ensure that every forwarder receives a tangible gain from the collaborative scheme (see Gansterer and Hartl, 2018 for examples). The lack of this perceived gain has been one of the main obstacles for the implementation of such schemes so far. Related to the previous point, game theoretical models, such as the ones in Hezarkhani et al. (2019) and Ciardiello et al. (2021), could be developed for this application. Moreover, qualitative studies may focus on barriers and enablers for such a cooperation and develop theoretical frameworks that support the deployment of this concept. Besides the complexity in cost/revenue sharing, the other potential issue is that FFs might get, by sharing capacity (trucks), business insights into competitors by somehow checking what and where other FFs are exporting. However, this aspect is limited in our approach since we rely on a neutral fleet that is not owned by any of the FFs. Also, the content of each ULD is not relevant for the fleet. Regarding the planning, an independent Information Technology (IT) company may act as central planner.

CRediT authorship contribution statement

Alessandro Bombelli: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Stefano Fazi:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Writing – original draft, Writing – review & editing.

Declaration of competing interest

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

Acknowledgments

Author A. B. would like to thank Felipe Delgado (Pontificia Universidad Católica de Chile) and Sebastian Birolini (University of Bergamo) for comments addressing an earlier version of the manuscript that improved its quality. We want to thank the Center for Information Technology of the University of Groningen for their support and for providing access to the Peregrine high performance computing cluster. Finally, we thank the Editor and three anonymous reviewers for their comments during the review process that improved the final outcome of the paper.

References

ACN, 2021. ACN Website. https://acn.nl/projecten/elink/?lang=en. Accessed: May 2021.
Adewumi, A.O., Adeleke, O.J., 2018. A survey of recent advances in vehicle routing problems. Int. J. Syst. Assur. Eng. Manag. 9, 155–172.
Amaruchkul, K., Cooper, W.L., Gupta, D., 2007. Single-leg air-cargo revenue management. Transp. Sci. 41, 457–469.
Angelelli, E., Archetti, C., Peirano, L., 2020. A matheuristic for the air transportation freight forwarder service problem. Comput. Oper. Res. 123, http://dx.doi.org/10.1016/j.cor.2020.105002, 105002.
Ankersmit, S., Rezaei, J., Tavasszy, L., 2014. The potential of horizontal collaboration in airport ground freight services. J. Air Transp. Manage. 40, 169–181.
Archetti, C., Peirano, L., 2020. Air intermodal freight transportation: The freight forwarder service problem. Omega 94, 102040.
Bartodziej, P., Derigs, U., Malcherek, D., Vogel, U., 2009. Models and algorithms for solving combined vehicle and crew scheduling problems with rest constraints: an application to road feeder service planning in air cargo transportation. OR Spectrum 31, 405–429.
Benavent, E., Landete, M., Mota, E., Tirado, G., 2015. The multiple vehicle pickup and delivery problem with LIFO constraints. European J. Oper. Res. 243, 752–762.
Berbeglia, G., Cordeau, J.F., Gribkovskaia, I., Laporte, G., 2007. Static pickup and delivery problems: a classification scheme and survey. Top 15, 1–31.
Berger, S., Bierwirth, C., 2010. Solutions to the request reassignment problem in collaborative carrier networks. Transp. Res. E 46, 627–638.
Braekers, K., Ramaekers, K., Van Nieuwenhuyse, I., 2016. The vehicle routing problem: State of the art classification and review. Comput. Ind. Eng. 99, 300–313.
Buso, F., 2017. Analyzing the Milkrun Service Process for Inbound Freight at Schiphol Airport with Queuing Theory and Simulation (Master's thesis). University of Mannheim.
Cherkesly, M., Desaulniers, G., Irnich, S., Laporte, G., 2016. Branch-price-and-cut algorithms for the pickup and delivery problem with time windows and multiple stacks. European J. Oper. Res. 250, 782–793.
Cherkesly, M., Desaulniers, G., Laporte, G., 2015. Branch-price-and-cut algorithms for the pickup and delivery problem with time windows and last-in-first-out loading. Transp. Sci. 49, 752–766.
Cheung, T., Wong, C., 2015. A system dynamics model for horizontal collaboration in air cargo transport services. Asia Pac. J. Adv. Bus. Soc. Stud. 1, 72–80.
Ciardiello, F., Genovese, A., Luo, S., Sgalambro, A., 2021. A game-theoretic multi-stakeholder model for cost allocation in urban consolidation centres. Ann. Oper. Res. 1–24.
Cordeau, J.F., 2006. A branch-and-cut algorithm for the dial-a-ride problem. Oper. Res. 54, 573–586.
Cordeau, J.F., Iori, M., Laporte, G., Salazar González, J.J., 2010. A branch-and-cut algorithm for the pickup and delivery traveling salesman problem with LIFO loading. Networks 55, 46–59.
Dondo, R., Cerdá, J., 2015. The heterogeneous vehicle routing and truck scheduling problem in a multi-door cross-dock system. Comput. Chem. Eng. 76, 42–62.
Drljača, M., 2017. Air cargo handling process. In: ZIRP 2017, International Conference on Traffic Development, Logistics & Sustainable Transport New Solutions and Innovations in Logistics and Transportation.

Feng, B., Li, Y., Shen, Z.J.M., 2015. Air cargo operations: Literature review and comparison with practices. Transp. Res. C 56, 263–280.
Gansterer, M., Hartl, R.F., 2018. Collaborative vehicle routing: a survey. European J. Oper. Res. 268, 1–12.
Gebhardt, S., Delgado, F., Angulo, G., 2015. Design of robust operation schedules for a freighter airline.
Hezarkhani, B., Slikker, M., Van Woensel, T., 2019. Gain-sharing in urban consolidation centers. European J. Oper. Res. 279, 380–392.
Huang, K., Chi, W., 2007. A Lagrangian relaxation based heuristic for the consolidation problem of airfreight forwarders. Transp. Res. C 15, 235–245.
Huang, K., Lee, Y.T., Xu, H., 2020. A routing and consolidation decision model for containerized air-land intermodal operations. Comput. Ind. Eng. 141, 106–299.
IATA, 2020. IATA Website. https://www.iata.org/en/programs/cargo/sustainability/benefits/. Accessed: April 2020.
Iori, M., Martello, S., 2010. Routing problems with loading constraints. Top 18, 4–27.
Khan, M.R., 2000. Business process reengineering of an air cargo handling process. Int. J. Prod. Econ. 63, 99–108.
Konur, D., Golias, M.M., 2013. Analysis of different approaches to cross-dock truck scheduling with truck arrival time uncertainty. Comput. Ind. Eng. 65, 663–672.
Ladier, A.L., Alpan, G., 2016. Cross-docking operations: Current research versus industry practice. Omega 62, 145–162.
Lai, M., Xue, W., Hu, Q., 2019. An ascending auction for freight forwarder collaboration in capacity sharing. Transp. Sci. 53, 1175–1195.
Lee, Y.H., Jung, J.W., Lee, K.M., 2006. Vehicle routing scheduling for cross-docking in the supply chain. Comput. Ind. Eng. 51, 247–256.
Leung, L.C., Cheung, W., Van Hai, Y., 2000. A framework for a logistics e-commerce community network: The Hong Kong air cargo industry. IEEE Trans. Syst. Man Cybern. A 30, 446–455.
Leung, L.C., Van Hui, Y., Chen, G., Wong, W.H., 2017. Aggregate-disaggregate approach to an airfreight forwarder's planning under uncertainty: a case study. J. Oper. Res. Soc. 68, 695–710.
Levitin, G., Abezgaouz, R., 2003. Optimal routing of multiple-load AGV subject to LIFO loading constraints. Comput. Oper. Res. 30, 397–410.
Lin, C., Choy, K.L., Ho, G.T., Chung, S.H., Lam, H., 2014. Survey of green vehicle routing problem: past and future trends. Expert Syst. Appl. 41, 1118–1138.
Liu, Y., Yin, M., Hansen, M., 2019. Economic costs of air cargo flight delays related to late package deliveries. Transp. Res. E 125, 388–401.
Lurkin, V., Schyns, M., 2015. The airline container loading problem with pickup and delivery. European J. Oper. Res. 244, 955–965.
Milkrun, 2021. ACN Website. https://acn.nl/milkrun/. Accessed: January 2021.
Nallian, 2021. Nallian website. https://www.nallian.com/applications/slot-booking/cs-slot-booking-bru. Accessed: January 2021.
Niu, B., Dai, Z., Zhuo, X., 2019. Co-opetition effect of promised-delivery-time sensitive demand on air cargo carriers' big data investment and demand signal sharing decisions. Transp. Res. E 123, 29–44.
Ou, J., Hsu, V.N., Li, C.L., 2010. Scheduling truck arrivals at an air cargo terminal. Prod. Oper. Manage. 19, 83–97.
Parragh, S.N., Doerner, K.F., Hartl, R.F., 2008. A survey on pickup and delivery problems. J. Betr. 58, 21–51.
Qin, C.r., Luo, L., You, Y., Xiao, Y.x., 2012. An optimization model of the single-leg air cargo space control based on Markov decision process. J. Appl. Math. 2012.
Rezaei, J., Hemmes, A., Tavasszy, L., 2017. Multi-criteria decision-making for complex bundling configurations in surface transportation of air freight. J. Air Transp. Manage. 61, 95–105.
Rijal, A., Bijvank, M., de Koster, R., 2019. Integrated scheduling and assignment of trucks at unit-load cross-dock terminals with mixed service mode dock doors. European J. Oper. Res. 278, 752–771.
Roesener, A.G., Barnes, J.W., 2016. An advanced tabu search approach to the dynamic airlift loading problem. Logist. Res. 9, 1–18.
Ropke, S., Pisinger, D., 2006. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transp. Sci. 40, 455–472.
Schaefer, A.J., Johnson, E.L., Kleywegt, A.J., Nemhauser, G.L., 2005. Airline crew scheduling under uncertainty. Transp. Sci. 39, 340–348.
Schiphol, 2021. Schiphol website. https://www.schiphol.nl/en/cargo/page/shaping-the-future-of-cargo-together/. Accessed: January 2021.
Shaw, P., 1997. A new local search algorithm providing high quality solutions to vehicle routing problems.
Simoni, M.D., Bujanovic, P., Boyles, S.D., Kutanoglu, E., 2018. Urban consolidation solutions for parcel delivery considering location, fleet and route choice. Case Stud. Transp. Policy 6, 112–124.
The loadstar, 2021. The loadstar website. https://theloadstar.com/data-sharing-is-nice-but-cooperation-is-key-to-ending-congestion-says-jan-de-rijk/. Accessed: January 2021.
Van Alebeek, S., Bombelli, A., 2021. Auction-based coopetition in the landside air cargo supply chain. J. Supply Chain Manage. Sci. 2, 3–18.
Vracken, R.E.M., 2020. Optimizing the Freight Handling in the Air Cargo Handling Chain: The Concept of Drop and Collect (Master's thesis). Eindhoven University of Technology.
Wu, Y., 2010. A dual-response forwarding approach for containerizing air cargoes under uncertainty, based on stochastic mixed 0-1 programming. European J. Oper. Res. 207, 152–164.
Xu, D., Zhang, C.W., Miao, Z., Cheung, R.K., 2014. A flow allocation strategy for routing over multiple flow classes with an application to air cargo terminals. Comput. Oper. Res. 51, 1–10.