Operations Research Forum (2023) 4:54
https://doi.org/10.1007/s43069-023-00232-5

RESEARCH

Integrating Micro-Depot Freight Transport in Existing Public Transport Services

Lena Hörsting[1] · Catherine Cleophas[1]

Received: 30 January 2023 / Accepted: 30 May 2023 / Published online: 8 July 2023

Abstract
As conventional last-mile transport contributes to traffic congestion and pollution, urban areas need new approaches to transporting freight. One promising idea is integrating freight deliveries with existing public transport infrastructures like light rail. However, this concept creates the challenge of offering a high service quality for passengers and freight. In this work, we consider a setting where freight originates from and is transhipped at several public transit stops that serve as micro-depots with a limited storage capacity. Furthermore, the system relies on shared vehicles, where a dedicated share of the capacity can be used to fasten freight containers or as a standing area for passengers. For this setting, we propose an optimisation model that integrates the tactical scheduling of transport services and the allocation of freight containers to those services. To solve realistically sized instances, we propose an adaptive large neighbourhood search heuristic. We use this heuristic to evaluate the system's sensitivity to the capacity of micro-depots and vehicles.

Keywords Transportation · Logistics · Shared transport · Scheduling · Adaptive large neighbourhood search

1 Introduction

Given close quarters and a high population density in urban areas, sustainably moving humans and freight on the "last mile" is challenging [1]. Both public transport and freight transport require space within cities' infrastructure and meet

This article is part of the Topical Collection on *Public Transport Optimization: From Theory to Practice*

✉ Lena Hörsting
lena.hoersting@bwl.uni-kiel.de

Catherine Cleophas
cleophas@bwl.uni-kiel.de

[1] Research Group Service Analytics, Christian-Albrechts-Universität zu Kiel, Westring 425, Kiel 24118, Schleswig-Holstein, Germany

similar expectations: Passengers expect short travel times and frequent departures, while freight customers expect quick and punctual deliveries. At the same time, citizens expect the municipality to limit air and noise pollution and traffic for a better quality of life.

Integrating freight transport into public transport may offer chances to improve both services: The option to exploit capacity overhangs profitably can justify financial investments in public transport while limiting the adverse ecological effects of freight transport. Capacity overhangs occur in on-demand [2] and in fixed transportation services [3]. Here, we consider the challenge of integrating passengers and freight on a fixed infrastructure, such as light rail, thereby further justifying the investment while exploiting the opportunity of short travel times compared to crowded streets [4]. We allow for a partially shared vehicle capacity, e.g., a dedicated area in each shared vehicle can either carry securely attached freight containers or serve as standing space for passengers. Although this concept currently faces legal restrictions in many European countries, it features in the pilot project LogIKtram in Karlsruhe, Germany [5].

Integrating passengers and freight must not disadvantage passengers to ensure the system's acceptance and keep public transport attractive [6]. The success of such systems depends on strategic decisions, such as the inclusion of public transport lines, vehicle capacity, and the capacity and positioning of depot stations [7, 8], as well as on operational decisions, such as how much freight to accept and which parcels to transport on which vehicle [9].

To support planning for such an integrated system and to evaluate its sensitivity to strategic decisions, we extend a model first presented in [10]. That contribution proposed a lexicographical model to create demand-oriented schedules and to assign freight containers to shared services given a single depot. The optimisation model prioritises passenger service by minimising the passengers' waiting time first and minimising the number of rejections and the delivery delay of cargo requests second. The extended model presented here lets freight originate and end at multiple *micro-depots* with a limited capacity. To solve realistically sized instances, we propose an adaptive large neighbourhood search (ALNS) heuristic as inspired by [11].

We evaluate the heuristic's solution performance and the system's sensitivity to the vehicle and micro-depot capacities by planning and processing a simulated day of operations. To that end, we generate artificial passenger and freight demand for various combinations of demand scenarios and network designs. Thus, the contribution of this paper is threefold:

- We formulate an extended problem allowing for the flexible pick-up and delivery of freight in a fixed public transport infrastructure.
- We extend ALNS operators for an ALNS-based approach to creating demand-oriented schedules and assign freight to transport services.
- We evaluate the system's sensitivity to strategic capacity decisions.

In the next section, we briefly review the state of integrated passenger and freight transport research before presenting an overview of related contributions featuring

ALNS. Subsequently, in Sect. 3, we describe the problem setting in further detail. In Sect. 4, we introduce the hierarchical optimisation model. Section 5 describes the proposed ALNS, specifically expounding on destroy and repair operators. We analyse the solution performance of the ALNS compared to simple scheduling rules and the mathematically optimal solution in Sect. 6. Section 7 analyses the resulting system's sensitivity to alternative combinations of the depot and vehicle capacity given by earlier strategic decisions. Finally, Sect. 8 summarises our findings and provides an agenda for further research in the domain.

2 State of the Art

Cooperative transport of freight and passengers becomes possible when the public transport system is not always fully utilised. Cavallaro and Nocera [7] provide a broad, concept-centric literature review focused on the idea. Hörsting and Cleophas [10] survey existing optimisation models considering the shared transportation of passengers and freight in an urban area on a fixed infrastructure. However, to provide a path to success for such integrated systems, decision-makers must find operational modes that ensure both passenger and freight service quality. In that regard, the research presented here is further motivated by a call-to-action stated in [12]: "model the impacts of [...] the projects [...] to estimate impacts on congestion, operations, and environmental outcomes".

As shown in [13], significant distance savings can result when public transport systems deliver freight to micro-depots, from where they can reach their final destination. The idea of these micro-depots motivates the extended model presented here, in which, at a variety of stops, both passengers and freight can leave and enter the vehicles.

As we propose, on the one hand, a prescriptive approach to schedule vehicles and allocate freight and, on the other hand, evaluate the outcomes for various scenarios, our research is related to several further contributions. For example, [14] evaluate scheduling decisions for cargo vehicles in a freight rail system based on a discrete event-based simulation. Taking a much broader view of the matter, [15] evaluate a range of urban logistics schemes in an agent-based simulation to estimate implications for sustainability and the schemes' attractiveness for stakeholders.

In the remainder of this section, we focus on two aspects that drive the model and the solution approach proposed in this paper: Firstly, we briefly summarise existing research on multi-depot transportation and relate it to the idea of micro-depots on a fixed infrastructure. Secondly, we provide some background for the adaptive large neighbourhood search algorithm we rely on to solve the scheduling problem.

2.1 Micro-Depot Transportation

Fragmented freight flows and inefficient delivery operations are two challenges in last-mile transportation that increase the need for consolidation [15]. At the same time, ecologically friendly transportation modes like cargo bikes and electronic trucks often offer a lower capacity and a shorter range than conventional trucks. This conundrum has sparked increasing interest in urban consolidation centres or micro-depots. Katsela et al. [16] evaluate the advantages and disadvantages of micro-hubs by analysing and comparing case studies. Planning a micro-depot network involves optimisation problems on various levels.

In recent research, [17] propose a method to locate distribution hubs in a metro network, while [18] consider location planning problems for multiple depots when cargo bikes take over the last part of freight transport. Furthermore, [19] review existing hub location problems and propose ideas to improve modelling. Delle Donne et al. [8] envision a system where freight enters and leaves the public transport system via a set of drop-in and drop-out stations. In that setting, the authors evaluate strategic decisions on which public transport lines to recruit for freight transport and where to locate drop-in and drop-out stations.

In vehicle routing, multiple depots are a prominent concept [20–23]. Multi-depot solutions also combine well with new logistics schemes like crowd shipping or mobile depots [24–26].

In passenger transport, approaches to scheduling focus on depots that describe the starting and ending point of the trip for each vehicle. For example, [27] consider a multi-depot vehicle scheduling problem where buses are assigned to given timetabled trips, whereas [28] focus on an integrated multi-depot vehicle and crew scheduling problem.

In this work, we consider a micro-depot network for shared passenger and freight transport. In that, we consider location and capacity decisions as given from strategic planning but evaluate their implications for integration success. Given the idea of fixed infrastructure and line plans, scheduling services and allocating freight to services overtakes routing.

2.2 Adaptive Large Neighbourhood Search (ALNS)

The *adaptive large neighbourhood search (ALNS)* is a meta-heuristic first proposed by [29] to solve a pickup and delivery problem with time windows. It extends the idea of *large neighbourhood search (LNS)* by allowing for a variety of destroy and repair operations as opposed to a single pair of one destroy and one repair operator. In each iteration, these destroy and repair operators are selected randomly according to weights that are updated based on different search strategies. For instance, operators leading to an improvement in the objective value are more likely to be selected in future iterations. The advantage of ALNS over LNS is the extended search space, which reduces the risk of getting stuck in local optima. Most ALNS algorithms include operators that aim for diversification and intensification. Diversification means the operator randomly searches a large part of the neighbourhood to find new solutions. Intensification implies that the operator searches purposefully to improve the quality of the current solution. For more information, we refer the reader to the definitions by [30].

Today, ALNS is applied to various large-scale routing and scheduling problems. For instance, for variants of the pickup and delivery problems [29, 31, 32], two-echelon vehicle routing problems [34, 33], or workforce routing and scheduling problems [35–37]. Windras Mara et al. [38] present a detailed survey of the recent development on ALNS.

Barrena et al. [39] introduce an ALNS heuristic to solve a rail rapid transit problem with dynamic passenger demand, minimising the passenger waiting time. They improve the results of a branch-and-bound algorithm in [40] by 26%. Similarly, [11] propose an ALNS to find a demand-oriented timetable, minimising the passenger congestion at stops. Their model includes limited vehicle capacities and multiple lines. Yuan et al. [41] use ALNS to design an urban network combining a fixed-line transport system and a demand-responsive transport system.

For this work, we are interested in applying ALNS to demand-oriented service scheduling problems. To that end, we adapt the heuristic of [11] to integrate the freight transport in Sect. 5.

3 Problem Setting

In the following, we explain the problem framework and underlying assumptions. We extend the model of [10] by considering multiple micro-depots to be located at selected stops and feature a limited storage capacity. Furthermore, we focus on the operational mode featuring shared vehicles, assuming a limited share of the vehicle capacity can be used to secure freight.

The setting features an existing public transit infrastructure with a fixed set of stops and connecting legs, e.g., a single line in a light-rail network. Each *station* consists of two *stops* with opposite directions. Furthermore, we assume that the stops of each station are close to each other, i.e., it is possible to walk from the first to the second stop of the same station in, at most, a few minutes. Such a network configuration is common in many European cities that are the inspiration for this work. The vehicles operate on a circular, bi-directional single route, i.e., they start at a joint *start stop* and turn around when arriving at the boundary station (see Fig. 1).

The expected passenger demand takes the form of quantity flows driven by the arrival and alighting rates per time period and stop. First, we consider the number of passengers arriving per stop and period. When a vehicle arrives at a stop, a fixed ratio of passengers alights. Since the network is bi-directional and each stop serves a dedicated direction, every passenger waiting at a stop wants to board the arriving vehicle.

Several stops are pre-defined *micro-depots*, where freight can be loaded and unloaded. We consider freight as *containers* carrying parcels with the same requirements, i.e., time and place of release, due time, and destination. Every container should be transported to its destination stop within a soft due time. Additionally, a hard due time per container sets the time of the latest acceptable arrival at the destination. We allow for the rejection of container requests to guarantee feasibility. Note that we assume that the service provider decides on the acceptance of containers before the start of the period so that freight customers can still find alternative means of transport. The micro-depots can only store

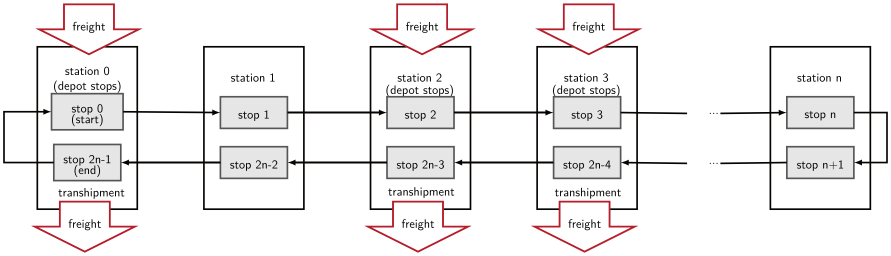


Fig. 1 Assumed network topology: a circular, bi-directional single route

a limited number of containers between release and pick-up. Containers do not require storage capacity after drop-off at their destination stop, as we assume they are picked up instantly or immediately forwarded to their final destination.

The vehicle capacity is also limited. Specifically, we assume that parts of the standing area for passengers are modified so that parcel containers can be attached there, i.e., we assume that passengers and cargo share a selected vehicle space. Again, the transport of passengers is prioritised. First, passengers alight and board the vehicle. If there is enough space left, the parcel containers are secured in the remainder of the shared area.

This work focuses on satisfying the demand for the passenger and freight service while integrating freight into the public transit system. For this reason, we consider an irregular time schedule with variable dwell times. Each *service* in the time schedule is defined by an individual vehicle's arrival and departure time per stop over the course of the considered horizon. All processes, like boarding and alighting passenger and loading and unloading containers, must be executed during the dwell time. Additionally, there must be a sufficient dwell time between the departure of two subsequent vehicles from the same stop, and no two vehicles must occupy the same platform simultaneously.

In practice, the planners usually determine the lines and the timetable before considering the vehicle routes and crew schedules [42]. Here, we disregard the challenges of rolling stock. Thus, we set an upper bound for the maximal number of services offered within the planning horizon instead of defining a specific fleet size. Nevertheless, given a time schedule, we can deduce the maximal number of simultaneously operating vehicles and, thus, the minimum required fleet size to operate the line. We aim to determine an irregular, demand-oriented time schedule and assign containers to operating services following the problem formulation detailed in the next section.

4 Mathematical Formulation

The problem considered here entails defining a time schedule $\mathcal{T}$ as a set of individual services. Since we consider demand-oriented scheduling, we allow for aperiodic services, i.e., vehicles may arrive and depart irregularly. We formally define a

Table 1 List of index sets

Notation	Set	Description
$\mathcal{S}$		Set of stops
$\mathcal{P}$		Set of periods in the planning horizon
$\mathcal{P}_s^{\text{dep}}$	$\subset \mathcal{P}$	Set of departure times from stop $s \in \mathcal{S}$
$\mathcal{P}_s^{\text{start}}$	$\subset \mathcal{P}$	Set of periods at which stop $s \in \mathcal{S}$ releases a container
$\mathcal{T}$		Set of services
$\mathcal{T}_{s,p}^{\text{dep}}$	$\subset \mathcal{T}$	Set of services that depart from stop $s \in \mathcal{S}$ at period $p \in \mathcal{P}$
$\mathcal{T}_{s,\leq p}^{\text{dep}}$	$\subset \mathcal{T}$	Set of services that depart from stop $s \in \mathcal{S}$ at period $p \in \mathcal{P}$ and earlier
$\mathcal{R}$		Set of containers
$\mathcal{R}_s$	$\subset \mathcal{R}$	Set of containers for which stop $s \in \mathcal{S}$ is included in the delivery path
$\mathcal{R}_s^{\text{start}}$	$\subset \mathcal{R}$	Set of containers with start at stop $s \in \mathcal{S}$
$\mathcal{R}_s^{\text{end}}$	$\subset \mathcal{R}$	Set of containers with stop $s \in \mathcal{S}$ as their destination
$\mathcal{R}_{s,\geq p}^{\text{start}}$	$\subset \mathcal{R}$	Set of containers that start at stop $s \in \mathcal{S}$ at period $p \in \mathcal{P}$ and later

schedule in Sect. 4.1. For a given schedule, we compute the passenger flows and the resulting passenger service quality as described in Sect. 4.2.

Additionally, the problem entails assigning cargo containers to vehicles as formalised in Sect. 4.3, while minimising cargo delay and rejection. In Sect. 4.4, we link both sub-problems through a global lexicographical objective function defined, where we minimise the passenger congestion first and afterward the delay and rejection rate of cargo requests. Tables 1, 2, and 3 list the notation for sets, variables, and parameters introduced in the following.

4.1 Time Schedule Formulation

We consider a discrete planning horizon, given as a set of time periods $\mathcal{P}$. Furthermore, we define a time schedule $\mathcal{T}$ as a set of individual services, where each

Table 2 List of variables

Notation	Set	Description
$d_{s,t}$	$\in \mathcal{P}$	Departure time of service $t \in \mathcal{T}$ from stop $s \in \mathcal{S}$
$a_{s,t}$	≥ 0	Arrival time of service $t \in \mathcal{T}$ at stop $s \in \mathcal{S}$
$h_{s,t}$	≥ 0	Dwell time of service $t \in \mathcal{T}$ at stop $s \in \mathcal{S}$
$w_{s,p+\tau_p^s}$	$\in \mathbb{N}$	Number of passengers at stop $s \in \mathcal{S}$ in the interval $[p, p + \tau_p^s]$ for period $p \in \mathcal{P}_s^{\text{dep}} \cup \{0\}$
$w_{s,p+\tau_p^s}^{\text{cum}}$	$\in \mathbb{N}$	Accumulated number of passengers at stop $s \in \mathcal{S}$ in the interval $[p, p + \tau_p^s]$ for period $p \in \mathcal{P}_s^{\text{dep}} \cup \{0\}$
$n_{s,t}$	$\in \mathbb{N}$	Number of passengers using service $t \in \mathcal{T}$ when departing from stop $s \in \mathcal{S}$
$b_{s,t}$	$\in \mathbb{N}$	Number of boarding passenger for service $t \in \mathcal{T}$ from stop $s \in \mathcal{S}$
$e_{s,t}$	$\in \mathbb{N}$	Number of alighting passenger from service $t \in \mathcal{T}$ at stop $s \in \mathcal{S}$
$y_{r,t}$	$\in \{0, 1\}$	Is one if the container $r \in \mathcal{R}$ assigned to service $t \in \mathcal{T}$, else zero
z_r	$\in \{0, 1\}$	Is one if the container $r \in \mathcal{R}$ accepted, else zero

Table 3 List of parameters

Notation	Set	Description
μ^{s_1,s_2}	≥ 0	Travel time between stop s_1 and stop $s_2 \in \mathcal{S}$
$\beta^{\min}$	≥ 0	Minimal dwell time at a stop
$\beta^{\max}$	≥ 0	Maximal dwell time at a stop
H	≥ 0	Minimal headway between two services
V	$\in \mathbb{N}$	Maximal number of services
τ_p^s	≥ 0	Duration between period $p \in \mathcal{P}$ to the next larger period in $\mathcal{P}_s^{\text{dep}}$
$N_{s,p}$	$\in \mathbb{N}$	Number of passengers arriving at stop $s \in \mathcal{S}$ at period $p \in \mathcal{P}$
$\lambda_{s,t}^{\text{alight}}$	$\in [0, 1]$	Ratio of passengers alighting service $t \in \mathcal{T}$ at stop $s \in \mathcal{S}$
C^{pas}	$\in \mathbb{N}$	Vehicle capacity primarily for passengers
β^{board}	≥ 0	Boarding time per passenger
β^{alight}	≥ 0	Alighting time per passenger
$\bar{h}_{s,t}$	≥ 0	(Fixed) dwell time of service $t \in \mathcal{T}$ at stop $s \in \mathcal{S}$
$T_{r,t}$	≥ 0	Delay of container $r \in \mathcal{R}$ transported according to service $t \in \mathcal{T}$
P	≥ 0	Penalty for rejection per container
Q_r	> 0	Space required to load a single container $r \in \mathcal{R}$
$\bar{n}_{s,t}$	$\in \mathbb{N}$	(Fixed) number of passengers using service $t \in \mathcal{T}$ when departing from stop $s \in \mathcal{S}$
C^{fre}	$\in \mathbb{N}$	Vehicle capacity for cargo only
C^s	$\in \mathbb{N}$	Micro-depot capacity of stop $s \in \mathcal{S}$
β^{load}	≥ 0	Loading time per container
β^{unld}	≥ 0	Unloading time per container
$T^{\max}$	≥ 0	Maximal delay per container

service $t \in \mathcal{T}$ must pass all stops in the bi-directional network, represented by the set $\mathcal{S}$. The number of services in $\mathcal{T}$ is limited by the parameter $V \in \mathbb{N}$. Each service $t \in \mathcal{T}$ consists of a departure time $d_{s,t} \in \mathcal{P}$, an arrival time $a_{s,t} \geq 0$, and a dwell time $h_{s,t} \geq 0$ per stop $s \in \mathcal{S}$. The parameter $\mu_{s_1,s_2} > 0$ gives the time-independent travel time between two stops s_1 and $s_2 \in \mathcal{S}$. A given parameter $\beta^{\min} \geq 0$ defines a *minimal dwell time*. Analogously, a *maximal dwell time* $\beta^{\max} \geq 0$ per service ensures a limited overall travel time. In a feasible solution, two vehicles cannot dwell at the same stop at the same time. A *minimal headway* $H \geq 0$ spaces the departure of two consecutive services from the same stop.

Formally, a feasible time schedule $\mathcal{T}$ solves the following decision model:

$$a_{s,t} = d_{s-1,t} + \mu_{s-1,s}, \quad \forall s \in \mathcal{S} \setminus \{0\}, t \in \mathcal{T}, \tag{1}$$

$$h_{s,t} = d_{s,t} - a_{s,t}, \quad \forall s \in \mathcal{S}, t \in \mathcal{T}, \tag{2}$$

$$h_{s,t} \geq \beta^{\min}, \quad \forall s \in \mathcal{S}, t \in \mathcal{T}, \tag{3}$$

$$h_{s,t} \leq \beta^{\max}, \quad \forall s \in \mathcal{S}, t \in \mathcal{T}, \tag{4}$$

$$a_{s,t} \geq d_{s,t-1}, \quad \forall s \in \mathcal{S}, t \in \mathcal{T} \setminus \{0\}, \tag{5}$$

$$d_{s,t} \geq d_{s,t-1} + H, \quad \forall s \in \mathcal{S}, t \in \mathcal{T} \setminus \{0\}, \tag{6}$$

$$|\mathcal{T}| \leq V, \tag{7}$$

$$d_{s,t} \in \mathcal{P}, h_{s,t}, a_{s,t} \geq 0, \quad \forall s \in \mathcal{S}, t \in \mathcal{T}. \tag{8}$$

Through constraints (1), each service's arrival time at the next stop results from adding up the departure time of the current stop and the travel time between both stops. Per stop, each service also provides a dwell time for the vehicle defined as in constraints (2). The minimal and maximal dwell are modelled in constraints (3) and (4). Constraints (5) set the order of the services, as a service's arrival cannot take place before the previous service's departure from the respective stop. Furthermore, constraints (6) ensure the headway between two consecutive departures from the same stop. Finally, constraint (7) limits the number of maximal services.

4.2 First Sub-Problem: Determine the Passenger Flow

We consider minimising freight transport's interference with passenger transport quality as a crucial success factor for their integration. To quantify the passenger service quality, we measure the average passenger waiting time, overall stops, and times resulting from the passenger flow that a specific schedule produces.

We let a mixed-integer program describe the passenger flow for a given time schedule $\mathcal{T}$. The parameter $N_{s,p} \in \mathbb{N}$ represents the new arriving passengers at stop $s \in \mathcal{S}$ and period $p \in \mathcal{P}$. Note that the number of passengers and containers at each stop only decreases at a service's departure. For this reason, we are particularly interested in the departure times, defined as set $\mathcal{P}_s^{\text{dep}} \subset \mathcal{P}$ for each stop $s \in \mathcal{S}$. Analogously, the set $\mathcal{T}_{s,p}^{\text{dep}} \subset \mathcal{T}$ includes the services that depart from stop $s \in \mathcal{S}$ at period p.

Additionally, the parameter $\tau_p^s > 0$ describes the duration between period $p \in \mathcal{P}_s^{\text{dep}}$ to the next larger period in $\mathcal{P}_s^{\text{dep}}$ for each stop $s \in \mathcal{S}$. The parameters $\beta^{\text{board}} \geq 0$ and $\beta^{\text{alight}} \geq 0$ give the required boarding and alighting time per passenger. Based on the schedule $\mathcal{T}$, the value of $\bar{h}_{s,t}$ represents the dwell time of service $t \in \mathcal{T}$ at stop $s \in \mathcal{S}$. The parameter C^{pas} is the vehicle capacity that can be used by passengers. The parameter $\lambda_{s,t}^{\text{alight}} \in [0, 1]$ represents the ratio of passengers alighting at stop $s \in \mathcal{S}$ from service $t \in \mathcal{T}$, where the boundary values indicate that none ($\lambda_{s,t}^{\text{alight}} = 0$) or all ($\lambda_{s,t}^{\text{alight}} = 1$) of the passengers alight.

The integer variable $w_{s,p+\tau_p^s}$ describes the number of passengers waiting during the time interval $[p, p + \tau_p^s]$ for period $p \in \mathcal{P}_s^{\text{dep}} \cup \{0\}$ at stop $s \in \mathcal{S}$. Analogously, the integer variable $w_{s,p+\tau_p^s}^{\text{cum}}$ represents the number of passengers accumulated over the time window $[p, p + \tau_p^s]$ for period p at stop s. The accumulation is important to record the waiting times per passenger. The integer variable $n_{s,t}$ represents the number of passengers travelling via service $t \in \mathcal{T}$ when departing from stop $s \in \mathcal{S}$. Finally, the integer variables $b_{s,t}$ and $e_{s,t}$ are the number of boarding and alighting passengers at stop s and for service t, respectively.

The overall sub-problem is defined as

$$\min A(\mathcal{T}) := \sum_{s \in \mathcal{S}} \sum_{p \in \mathcal{P}_s^{\text{dep}} \cup \{0\}} \frac{w^{\text{cum}}_{s,p+\tau_p^s}}{|\mathcal{S}| \cdot |\mathcal{P}|}, \tag{9}$$

$$\text{s.t. } w_{s,p+\tau_p^s} = w_{s,p} - \sum_{t \in \mathcal{T}^{\text{dep}}_{p+\tau_p^s,s}} b_{s,t} + \sum_{\tau=0}^{\tau_p^s-1} N_{s,p+\tau}, \qquad \forall p \in \mathcal{P}_s^{\text{dep}} \cup \{0\}, s \in \mathcal{S}, \tag{10}$$

$$w^{\text{cum}}_{s,p+\tau_p^s} = \tau_p^s \cdot w_{s,p} - \sum_{t \in \mathcal{T}^{\text{dep}}_{p+\tau_p^s,s}} b_{s,t} + \sum_{\tau=0}^{\tau_p^s-1} (\tau_p^s - \tau) \cdot N_{s,p+\tau}, \qquad \forall p \in \mathcal{P}_s^{\text{dep}} \cup \{0\}, s \in \mathcal{S}, \tag{11}$$

$$n_{s,t} = n_{s-1,t} + b_{s,t} - e_{s,t}, \qquad \forall s \in \mathcal{S} \setminus \{0\}, t \in \mathcal{T}, \tag{12}$$

$$n_{0,t} = b_{0,t}, \qquad \forall t \in \mathcal{T}, \tag{13}$$

$$e_{s,t} \leq \lambda^{\text{alight}}_{s,t} \cdot n_{s-1,t}, \qquad \forall s \in \mathcal{S} \setminus \{0\}, t \in \mathcal{T}, \tag{14}$$

$$e_{s,t} \geq \lambda^{\text{alight}}_{s,t} \cdot n_{s-1,t} - 1, \qquad \forall s \in \mathcal{S} \setminus \{0\}, t \in \mathcal{T}, \tag{15}$$

$$n_{s,t} \leq C^{\text{pas}}, \qquad \forall s \in \mathcal{S}, t \in \mathcal{T}, \tag{16}$$

$$\beta^{\text{board}} \cdot b_{s,t} + \beta^{\text{alight}} \cdot e_{s,t} \leq \bar{h}_{s,t}, \qquad \forall s \in \mathcal{S}, t \in \mathcal{T}, \tag{17}$$

$$w_{s,p}, w^{\text{cum}}_{s,p} \in \mathbb{N} \qquad \forall p \in \mathcal{P}_s^{\text{dep}} \cup \{0\}, s \in \mathcal{S}, \tag{18}$$

$$n_{s,t}, b_{s,t}, e_{s,t} \in \mathbb{N} \qquad \forall s \in \mathcal{S}, t \in \mathcal{T}. \tag{19}$$

The objective function (9) minimises the average passenger waiting time. Constraints (10) and (11) determine the absolute and accumulated number of waiting passengers. The number of passengers travelling on a service is represented by constraints (12). Constraints (13) set the number of initially boarding passengers to the number of initially waiting passengers. Constraints (14) and (15) consider the number of passengers who alight a service at a particular stop. Constraints (16) limit the number of onboard passengers to the available vehicle capacity. Constraints (17) model the required service times for passenger transportation and limit it to the overall dwell time per stop and service.

4.3 Second Sub-Problem: Allocate Containers in a Given Schedule

Given a schedule that maximises passenger service quality, we want to reward freight allocations that produce an acceptable freight service quality. Here, we measure freight service quality via delays and rejection penalties.

We introduce several index sets to model the transport of parcel containers, represented by the set $\mathcal{R}$. The set $\mathcal{T}_r$ contains all services suitable for transporting container $r \in \mathcal{R}$, i.e., ensuring that the service's departure does not occur before the container is released and loaded. The set $\mathcal{T}^{\mathrm{dep}}_{s,\leq p}$ includes all services that depart stop $s \in \mathcal{S}$ at period $p \in \mathcal{P}$ and earlier. The set $\mathcal{R}_s$ includes all containers for which stop $s \in \mathcal{S}$ is included in the delivery path. The set $\mathcal{R}^{\mathrm{start}}_s \subset \mathcal{R}$ ($\mathcal{R}^{\mathrm{end}}_s \subset \mathcal{R}$) includes all containers with start (destination) $s \in \mathcal{S}$. The set $\mathcal{R}^{\mathrm{start}}_{s,\geq p}$ includes the containers released at period $p \in \mathcal{P}$ and later. The set $\mathcal{P}^{\mathrm{start}}_s$ includes all periods at which stop $s \in \mathcal{S}$ releases a container.

The binary decision variable $y_{r,t}$ models whether container $r \in \mathcal{R}$ is assigned to service $t \in \mathcal{T}$ ($y_{r,t} = 1$) or not ($y_{r,t} = 0$). Furthermore, the binary variable z_r decides whether a container $r \in \mathcal{R}$ is accepted ($z_r = 1$) or rejected ($z_r = 0$).

The parameters $\beta^{\mathrm{load}} \geq 0$ and $\beta^{\mathrm{unld}} \geq 0$ represent the required time for loading and unloading a container. Furthermore, the value of $\bar{h}_{s,t}$ denotes the dwell time of service $t \in \mathcal{T}$ at stop $s \in \mathcal{S}$. The value $T_{r,t}$ presents the delay of containers $r \in \mathcal{R}$ if transported via service $t \in \mathcal{T}$. Furthermore, the delay of a container must not exceed the parameter $T^{\max} \geq 0$. Each rejected container is penalised with a parameter $P \geq 0$
.

As stated before, the parameter $C^{\mathrm{pas}} > 0$ is the total vehicle capacity that can be used by passengers. However, the vehicle's standing area can alternatively hold parcel containers; its size is represented by the parameter $C^{\mathrm{fre}} \geq 0$. Thus, $C^{\mathrm{fre}} \leq C^{\mathrm{pas}}$ always holds. The parameter $Q_r > 0$ gives the space required to transport a single container $r \in \mathcal{R}$ in relation to the space required by one passenger. Figure 2 illustrates the relation of the parameters C^{pas}, C^{fre} and Q_r, for $r \in \mathcal{R}$. The number of passengers $\bar{n}_{s,t}$ travelling on service $t \in \mathcal{T}$ by departure from stop $s \in \mathcal{S}$ results from

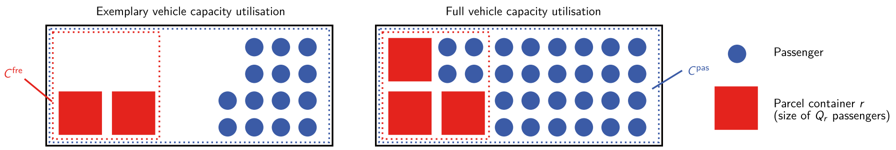

Exemplary vehicle capacity utilisation
Full vehicle capacity utilisation
C^{fre}
C^{pas}
Passenger
Parcel container r
(size of Q_r passengers)

Fig. 2 Illustration for the concept of shared vehicle capacity

solving the first sub-problem (cf. Sect. 4.2). Additionally, the parameter $C^s \geq 0$ describes the freight storage capacity of the depot stop $s \in \mathcal{S}$.

Thus, allocating containers to vehicles in a given schedule entails solving the following second sub-problem:

$$\min B(\mathcal{T}) := \sum_{r \in \mathcal{R}} \left(\sum_{t \in \mathcal{T}} \frac{T_{r,t} \cdot y_{r,t}}{|\mathcal{R}|} + P \cdot (1 - z_r) \right) \tag{20}$$

$$\text{s.t.} \sum_{t \in \mathcal{T}_r} y_{r,t} = z_r, \quad \forall r \in \mathcal{R}, \tag{21}$$

$$\sum_{t \in \mathcal{T} \setminus \mathcal{T}_r} y_{r,t} = 0, \quad \forall r \in \mathcal{R}, \tag{22}$$

$$\sum_{r \in \mathcal{R}_s} Q_r \cdot y_{r,t} \leq C^{\text{pas}} - \bar{n}_{s,t}, \quad \forall s \in \mathcal{S}, t \in \mathcal{T}, \tag{23}$$

$$\sum_{r \in \mathcal{R}_s} Q_r \cdot y_{r,t} \leq C^{\text{fre}}, \quad \forall s \in \mathcal{S}, t \in \mathcal{T}, \tag{24}$$

$$\sum_{r \in \mathcal{R}^{\text{start}}_{s,\geq p}} \left(z_r - \sum_{t \in \mathcal{T}^{\text{dep}}_{s,\leq p}} y_{r,t} \right) \leq C^s, \quad \forall p \in \mathcal{P}^{\text{start}}_s, s \in \mathcal{S}, \tag{25}$$

$$\sum_{r \in \mathcal{R}^{\text{start}}_s} \beta^{\text{load}} \cdot y_{r,t} + \sum_{r \in \mathcal{R}^{\text{end}}_s} \beta^{\text{unld}} \cdot y_{r,t} \leq \bar{h}_{s,t}, \quad \forall s \in \mathcal{S}, t \in \mathcal{T}, \tag{26}$$

$$T_{r,t} \leq T^{\max} \cdot y_{r,t}, \quad \forall r \in \mathcal{R}, t \in \mathcal{T}, \tag{27}$$

$$y_{r,t}, z_r \in \{0, 1\}, \quad \forall r \in \mathcal{R}, t \in \mathcal{T}. \tag{28}$$

Objective function (20) minimises the average cargo delay and the total rejection penalty. Constraints (21) ensure the delivery of each container via exactly one suitable service. Conversely, a service must not be selected if its departure at a stop is set before the containers are released (cf. constraints (22)). Constraints (23) guarantee the restriction to the left-over capacity after the passengers boarded. Simultaneously, constraints (24) limit the used space to the dedicated cargo capacity of the vehicle. Constraints (25) model a limited stop capacity to store accepted containers in micro-depots between their release and pick-up by the assigned service. Rejected containers do not require storage. Constraints (26) model the required service times for cargo transportation. Constraints (27) enforce a maximal delay per container.

4.4 Global Objective Function

We rely on a lexicographical objective as previously implemented in [10] to optimise passenger and freight service quality. We prioritise passenger transport, such that the primary objective is to reduce the average waiting time $A(\mathcal{T})$ accumulated over all periods and stops (cf. Sect. 4.2). The secondary objective minimises the penalty $B(\mathcal{T}, y, z)$ for average container delivery delay or container rejection (cf. Sect. 4.3). Thus, the global objective function is

$$\text{lex}\min_{\mathcal{T},y,z}\{A(\mathcal{T}), B(\mathcal{T}, y, z)\}, \tag{29}$$

where $\mathcal{T}$ is a feasible time schedule according to Sect. 4.1, and the assignment variable $y \in \{0,1\}^{\mathcal{R}\times\mathcal{T}}$ and the acceptance variable $z \in \{0,1\}^{\mathcal{R}}$ satisfy the restrictions stated in Sect. 4.3.

5 Adaptive Large Neighbourhood Search (ALNS)

Hörsting and Cleophas [10] show that solving a simpler version of the problem considered here to optimality is computationally expensive. As we aim to support planning and analysis for instances of realistic size in terms of the number of vehicles, stops, and periods, we propose to solve the problem via an ALNS heuristic. In that, we adapt the solution approach of [11] by integrating cargo transport. To that end, we propose iteratively optimising the cargo allocation and extending the ALNS by adding an operator that accounts for cargo service quality.

Figure 3 illustrates the ALNS procedure for the integrated cargo and passenger scheduling problem. The heuristic starts by constructing an initial time schedule (see Sect. 5.1), which it considers as the *current schedule*. Subsequently, it applies firstly a destroy operator (Sect. 5.2) and secondly a repair operator (Sect. 5.3). It selects operators based on probabilities given by the current weights. Applying the operators results in a new *candidate schedule*. The heuristic evaluates the passenger service quality for this candidate solution by solving the passenger flow problem. After a given number of warm-up iterations and if the candidate solution produces a passenger service quality that is at least equal to that in the current solution, the heuristic computes the optimal cargo assignment, i.e., solving the sub-problem introduced in Sect. 4.3. If the overall objective value improves, the candidate solution becomes the *new best solution*. Finally, the heuristic updates the operator weights and decides whether the candidate solution becomes the *new current solution* based on an acceptance criterion. This procedure repeats until a maximum number of iterations, a maximum number of iterations without improvement, or a maximal running time is exceeded. In the following, we discuss each step in more detail.

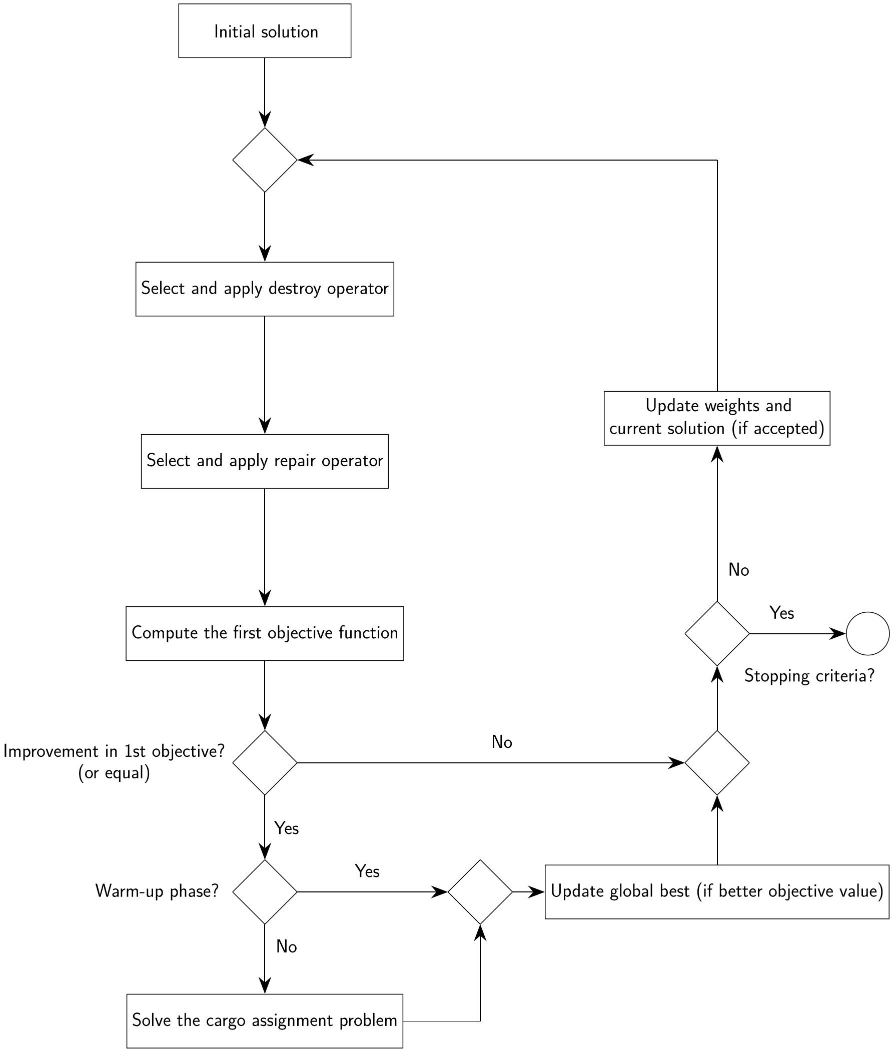

Initial solution
Select and apply destroy operator
Select and apply repair operator
Compute the first objective function
Improvement in 1st objective? (or equal)
No
Yes
Warm-up phase?
Yes
No
Solve the cargo assignment problem
Update global best (if better objective value)
Stopping criteria?
Yes
No
Update weights and current solution (if accepted)

Fig. 3 Adaptive large neighbourhood search for the integrated cargo and passenger scheduling problem

5.1 Initialisation

The ALNS algorithm creates an initial schedule before searching for improvements via destroy and repair operations. Specifically, the initialisation constructs a time schedule with regularly spaced departures and fixed dwell times at each stop. Figure 4 illustrates an exemplary initial schedule.

First, the initialisation fixes the latest feasible service for an initial constant dwell time h and stops $|\mathcal{S}|$. This service departs from the last stop at the last possible period, i.e., $|\mathcal{P}| - 1$. Thus, the last possible arrival at the starting stop depends on the dwell time h and travel time $\mu_{s-1,s}$ between stops $s-1$ and s;

$$A(h) := |\mathcal{P}| - 1 - |\mathcal{S}| \cdot h - \sum_{s \in \mathcal{S} \setminus \{0\}} \mu_{s-1,s}. \tag{30}$$

If $A(h)$ is smaller than zero, the initial dwell time h is too large to ensure a feasible schedule. In that case, the initialisation iteratively decrements the dwell time and computes the new resulting $A(h)$.

As the initial schedule features a constant headway, the initialisation defines this headway between two departures from the same stop for V services as

$$H(h, V) := \frac{A(h)}{V-1}. \tag{31}$$

If $\lfloor H(h, V) \rfloor$ is smaller than the given minimal headway time or the dwell time per stop, the initialisation procedure iteratively updates $V \rightarrow V-1$ until a feasible time schedule with a sufficient headway $H(h, V)$ results.

Subsequently, the initialisation constructs the schedule by fixing the last service and adding the remaining services with headway $H(h, V)$. This procedure rounds up the departure times to ensure they are feasible, i.e., included in the set $\mathcal{P}$. Accordingly, service $t \in \{1, \ldots, V\}$ ends at

$$D(h, V, t) := \lceil |\mathcal{P}| - 1 - H(h, V) \cdot (V - t) \rceil. \tag{32}$$

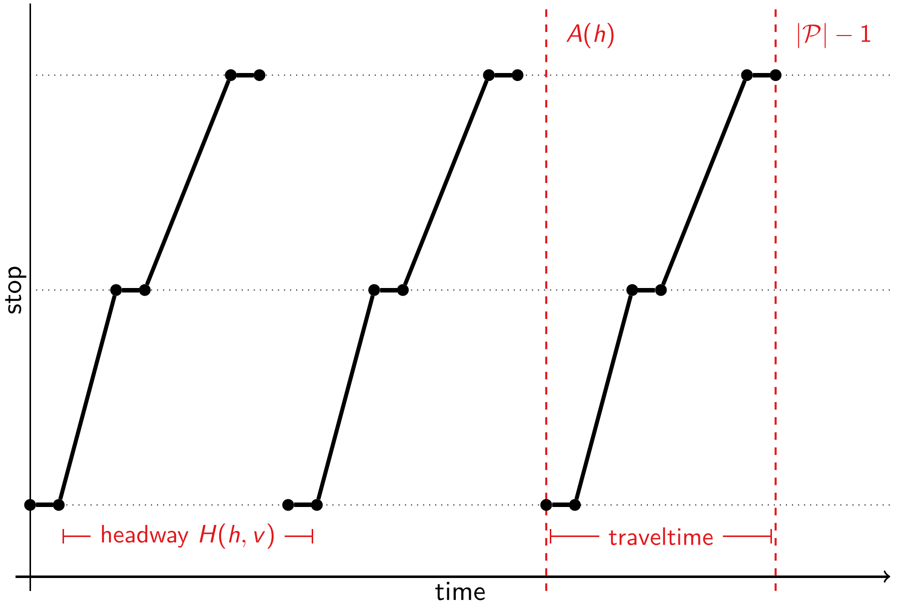


Fig. 4 Initial time schedule

The remaining departures at previous stops are added according to the given dwell time *h* and the travel time between two stops.

5.2 Destroy Operators

The heuristic proposed here features three destroy operators as illustrated in Fig. 5. The operator *random deletion and shift* supports diversification, while the operators *passenger-driven deletion* and *freight-driven deletion* drive intensification. Figure 5 illustrates how the destroy operators' steps change a given schedule.

5.2.1 Random Deletion and Shift

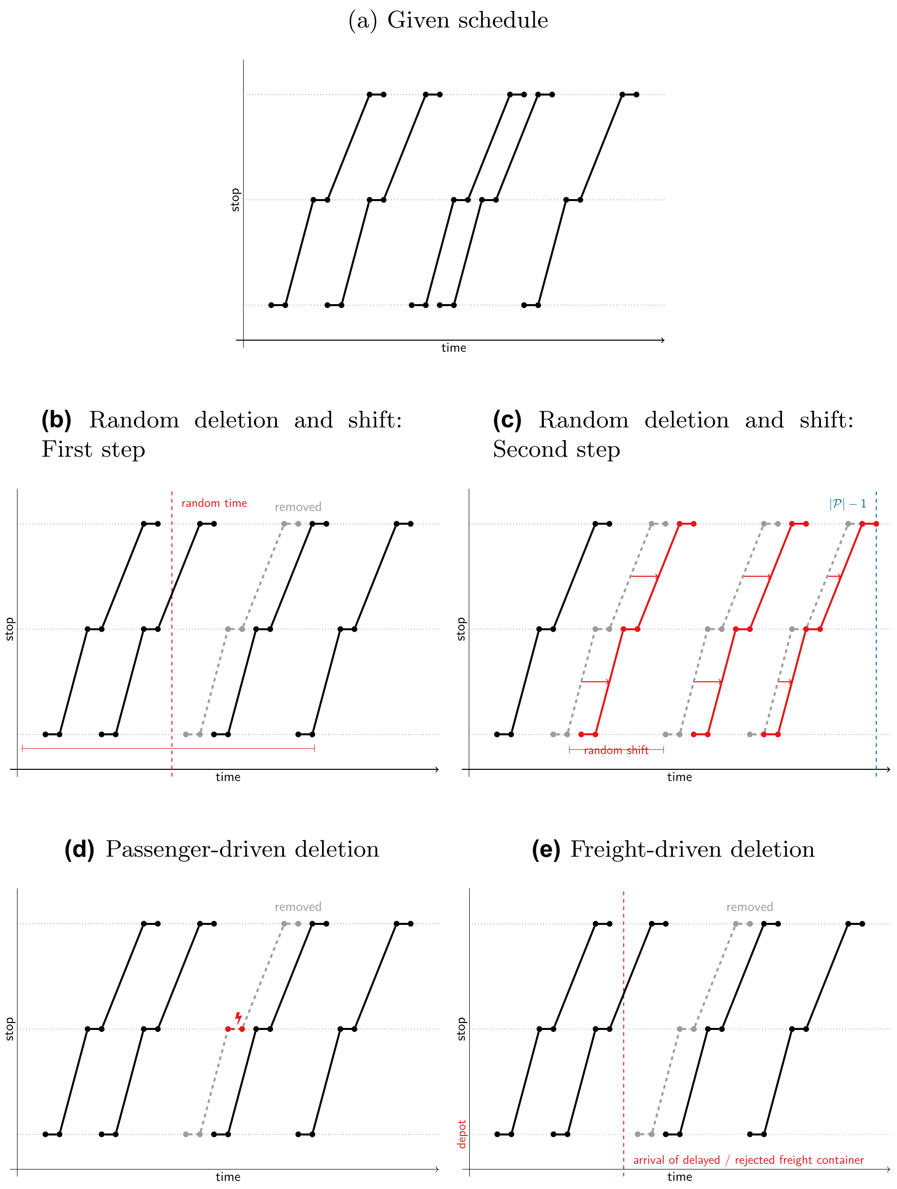
This operator features of two consecutive steps to create space for new services, a random deletion followed by a shift of the remaining service. The first step (see Fig. 5b) randomly selects a period between the first period and the last departure from the first stop. It removes the next service leaving the first stop after the selected time period from the schedule. The second step (see Fig. 5c) aims to distribute the gained space across the whole time window. To achieve this, it shifts an either all preceding or subsequent services in a random direction (*left* or *right*). The intensity of the shift is also randomly chosen to fall between zero and the size of the gap gained through the first step. If the direction of the shift is left (*right*), it shifts the service released after (*before*) the previously deleted service and all earlier (*subsequently*) released services. Due to the limited time window, shifting the first or last service may create an infeasible schedule. Therefore, the operator only shifts the boundary services as far as possible while maintaining feasibility. Then, it iteratively shifts the next services following the same approach. The second step is omitted if no shift creates a feasible schedule.

5.2.2 Passenger-Driven Deletion

This operator aims to improve passenger service quality by deleting low-quality services. To that end, it either deletes a service with insufficient dwell times for all passenger processes (see Fig. 5d) or, if that does not exist, deletes the service expected to carry the lowest accumulated number of passengers. To support the insertion of a new service by the following repair operator (see Sect. 5.3), the destroy operator stores and optionally adapts the pattern of dwell times of the deleted service. As illustrated in Fig. 6, it adapts the dwell times by increasing it for all stops with insufficient for the boarding and alighting process.

5.2.3 Freight-Driven Deletion

This operator deletes services of poor freight service quality. First, if the current solution includes rejected containers, the operator randomly selects one of those and deletes the service that departs after its arrival. Secondly, if the current solution includes delayed containers, it deletes the service departing after the release



(a) Given schedule

(b) Random deletion and shift: First step

(c) Random deletion and shift: Second step

(d) Passenger-driven deletion

(e) Freight-driven deletion

Fig. 5 Destroy operators

of the most severely delayed container. If the current solution plans for all containers to be delivered punctually, the operator returns the current solution without changes. Again, it saves the dwell times for each stop of the deleted service. If the increment of the dwell time at a particular stop allows for loading or unloading an additional freight container, it adapts the pattern accordingly (see Fig. 6).

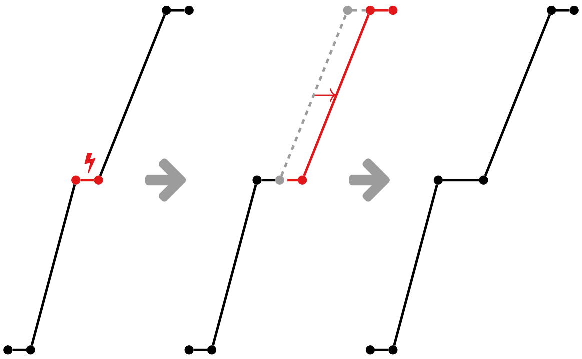


Fig. 6 Passenger and freight-driven deletion: adapt the dwell times for the new service

Consequentially, for a feasible current schedule, the destroy operators guarantee a feasible output schedule. Repair operators use the resulting gaps between services to introduce new services.

5.3 Repair Operators

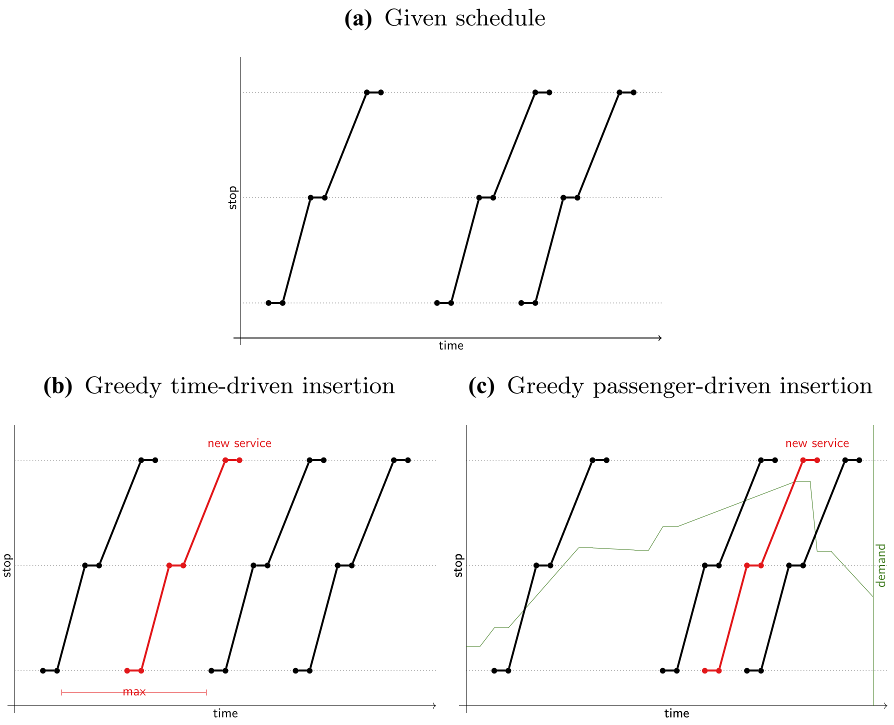
Following the destroy operator, the heuristic randomly selects and applies one of two possible repair operators based on the given weights. Figure 7 illustrates how each repair operator adjusts the schedule. Each repair operator inserts a new service according to certain criteria. The new services follow either the default or an adapted dwell time pattern, depending on the previously applied destroy operator.

5.3.1 Greedy Time-Driven

The first repair operator (see Fig. 7b) selects the largest gap between the departure time of one service and the arrival of the next one in the given time schedule. Considering $A(s, t)$ as the arrival time and $D(s, t)$ as the departure time of service t at stop s, it computes this gap as

$$\max_{s \in \mathcal{S}, t \in \mathcal{T}} I(s,t) = \begin{cases} A(s,t), & \text{if } t \text{ is the first service,} \\ |\mathcal{P}| - 1 - D(s,t), & \text{if } t \text{ is the last service,} \\ A(s,t+1) - D(s,t), & \text{else.} \end{cases} \tag{33}$$

The operator inserts a new service at the centre of the gap if this still allows for the required minimal headway. Then, it selects the now largest gap and repeats the insertion until either reaching the maximal number of services or violating the headway requirement.


(a) Given schedule

(b) Greedy time-driven insertion

(c) Greedy passenger-driven insertion

Fig. 7 Repair operators

5.3.2 Greedy Passenger-Driven

The second repair operator (see Fig. 7c) selects a gap for service insertion based on the expected effect on passenger service quality. To that end, the operator first identifies all time points $p \in \mathcal{P}$ where a service could be feasibly inserted. For each p, the operator ignores capacity and time limitations and computes the savings from picking up passengers who are currently waiting via the new service released at period p at the depot, t^{new}:

$$S(p) := \sum_{s \in \mathcal{S}} w_{s, D(s, t^{\text{new}}) - 1} \cdot (D(s, t^{\text{new}}) - D(s, t^{\text{new}} + 1)), \tag{34}$$

Then, the operator selects the time to insert the service to maximise the potential savings:

$$S(p^*) := \max_{p \in \mathcal{P}} S(p) \tag{35}$$

This operator also repeats insertions until reaching the maximum number of services or violating the headway requirements. Note that the operator computes the number of waiting passengers after each insertion by re-solving the passenger flow problem.

5.4 Search Strategy and Updating Weights

In the proposed ALNS, a simple scheme continuously updates the weights assigned to the two randomly applied repair operators. First, the initialisation assigns a uniform weight ρ_i to each repair operator i. Then, the probability of selecting the repair operator i is defined as

$$\theta_i = \frac{\rho_i}{\sum_{\rho_k \in \Omega^+} \rho_k}, \tag{36}$$

where Ω^+ is the set of all repair operators. Thus, the larger the weight ρ_i, the higher the probability of selecting operator i. The weights of destroy operators are computed analogously with a set Ω^- including all destroy operators.

Whenever a new candidate solution results from applying the destroy and repair operators, we define the new weights of the operator that contributed to creating that candidate solution as

$$\Psi := \begin{cases} \omega_1, & \text{if the new solution is new global best,} \\ \omega_2, & \text{if the new solution is better than the current solution,} \\ \omega_3, & \text{if the new solution satisfies the acceptance criterion,} \\ \omega_4, & \text{otherwise.} \end{cases} \tag{37}$$

Since the algorithm favours operators resulting in an improvement, we assume the order $\omega_1 \geq \omega_2 \geq \omega_3 \geq \omega_4$.

Furthermore, an *acceptance criterion* decides whether the current solution should reflect the newly found candidate solution. Thereby, the acceptance criterion sets the searching strategy through the neighbourhood. The *hill climbing (HC)* criterion accepts only solutions that are as good as or better than the current best, i.e., a candidate solution x is accepted if

$$f(x) \leq f(x^*), \tag{38}$$

where x^* is the best solution observed so far and f is the objective function.

Following the recommendation by [43], we also implement a linear record-to-record travel (RRT). The RRT criterion accepts a new solution if the relative difference to the global best is less than a continuously updated threshold. First, we define an initial threshold T^{start} and a smaller end threshold value T^{end}. A candidate solution x is accepted if

$$f(x) - f(x^*) \leq T. \tag{39}$$

At the end of each iteration, we linearly update the threshold value by

$$T \leftarrow \max\{T - \frac{T^{\text{start}} - T^{\text{end}}}{I^{\max}}, T^{\text{end}}\}, \tag{40}$$

where $I^{\max}$ is the maximal number of iterations.

Finally, the algorithm updates the weight as

$$\rho_a = \lambda \rho_a + (1 - \lambda)\Psi, \tag{41}$$

where the *decay parameter* $\lambda \in [0, 1]$ controls the sensitivity to changes. If λ is close to one, the weighting distribution only changes slowly, whereas a value close to zero highly rewards those operators that lead to improvement or acceptance in past iterations.

6 Performance Analysis

Before evaluating the sensitivity of the resulting system on instances of realistic size, we validate the performance of the ALNS versus the outcome from solving the problem to optimality. To this end, we implement the mixed-integer program given in Sect. 4 in Gurobi (v9.0.1) and limit the runtime to 9 h for both objectives. We implement the ALNS from Sect. 5 in Python 3 based on the open-source package *alns (v4.1.0)*. The hill climbing (HC) and record-to-record-travel (RRT) acceptance criterion rely on a starting threshold of 25% and an ending threshold of 1% of the initial first objective value. The algorithm starts with at most 100 warm-up iterations for both criteria, followed by at most 1900 regular iterations. Additionally, the heuristic stops after 500 iterations without an objective value improvement or after a maximal running time of 1 h. All computations are run on an INTEL Sandy Bridge processor with 2.6 GHz.

As performance indicators, we consider the global runtime and the runtime per objective function. Additionally, we consider the average relative gap to the best value found and the average relative optimality gap, defined as

$$\frac{\text{solution value} - \text{best value}}{\text{solution value}} \quad \text{and} \quad \frac{\text{lower bound} - \text{solution value}}{\text{solution value}}. \tag{42}$$

The source code underlying these results is publicly available at [44]. Furthermore, all instances and results presented in the following are publicly available under the same link.

For the performance analysis, we evaluate five problem settings with 60, 70, 80, 90, and 100 periods. We choose small instances to solve the problem to optimality within an acceptable runtime. We create 10 stochastic instances for each setting by randomly drawing the number of released containers per stop from a uniform probability distribution. Additionally, we consider ten instances of an exemplary large-scale scenario with 2880 periods. The scenario, termed *uniform freight flow, all-depots*, is introduced in more detail in Sect. 7. Due to their size, these instances are not solvable to optimality; therefore, we exclude a comparison to the solver.

Table 4 shows the average runtime and solution quality for the varying number of periods. Noticeably, the solver runtime rises exponentially with increasing periods. The solver cannot solve the problem to optimality within the given time limitations of 9 h per objective for more than 100 periods.

The overall results in terms of the objective values and runtime of attempting to solve the problem to optimality are similar to those of solving the predecessor model introduced in [10] under the equivalent computational settings. That contribution compared large-scale instances with 180 periods and small-scale instances with 60 periods. For the small-scale instances, the solver found an optimal solution for the majority of instances; only instances with a high number of parcel containers could not be solved to optimality in the second objective. None of the large-scale instances could be solved to optimality, remaining with optimality gaps between 65 and 99% for the first objective function. Hence, extending the model to allow for multiple micro-depots with limited and stop-dependent storage capacities seems to have no significant effect on the problem's computational cost. Both models show that the scale of the planning horizon is the primary driver of runtime and solving the problem for realistic horizons calls for efficient solution approaches like the ALNS presented in this paper.

To evaluate the heuristics' performance, we first consider the results for instances with at most 100 periods. In these instances, both ALNS variants still find a close-to-best solution with a gap of 2 to 5% to the best solution found by the solver in the first objective. The runtimes are at most 14 min for HC and at most 2 min for RRT. The difference in runtime mostly stems from HC solving the MIP for the second objective function more frequently. As described in Sect. 5, it only pursues the

Table 4 Average runtime and solution quality for a varying number of periods. The column *time (sec.)* includes the average optimisation time. Additionally, we consider the average *time*, the average relative *gap* to the best value found *(best)*, and the average relative optimality *gap* to the best lower bound found *(opt)* for both objective functions individually

Periods			**1st objective**			**2nd objective**		
		Time[a]	**Time**[a]	**Gap**	**Gap**	**Time**[a]	**Gap**	**Gap**
	Method	**(sec.)**	**(sec.)**	**(best)**	**(opt)**	**(sec.)**	**(best)**	**(opt)**
60	Solver	1853.4	167.0	0.00	0.00	444.1	0.00	0.00
60	ALNS HC	509.3	12.2	0.03	0.03	634.9	0.11	0.11
60	ALNS RRT	42.3	12.2	0.04	0.04	5.0	0.11	0.11
70	Solver	11,655.2	791.6	0.00	0.00	1688.5	0.00	0.00
70	ALNS HC	318.9	12.0	0.03	0.03	388.3	0.08	0.08
70	ALNS RRT	65.7	15.2	0.03	0.03	22.2	0.09	0.09
80	Solver	75,629.5	14,060.3	0.00	0.00	19,125.5	0.00	0.02
80	ALNS HC	822.8	12.7	0.05	0.05	577.9	0.05	0.07
80	ALNS RRT	87.4	14.9	0.04	0.04	40.4	0.05	0.07
90	Solver	102,163.0	27,431.5	0.00	0.07	22,730.7	0.00	0.03
90	ALNS HC	380.4	11.7	0.03	0.09	424.9	0.04	0.07
90	ALNS RRT	59.0	12.2	0.03	0.10	15.9	0.04	0.07
100	Solver	102,318.1	32,404.1	0.00	0.17	9779.8	0.00	0.01
100	ALNS HC	534.4	11.4	0.02	0.19	415.2	0.05	0.07
100	ALNS RRT	76.7	17.3	0.02	0.19	22.3	0.05	0.06
2880	ALNS HC	2822.8	617.8	0.00	-	691.4	0.18	-
2880	ALNS RRT	1432.7	478.2	0.03	-	65.8	0.06	-

[a] Runtime limit for solver set to of 32,400 s (9 h) per objective

second objective if the first objective of the candidate solution is at least as good as the current solution. Additionally, HC often includes more iterations because of the improvement stop criterion.

Different results emerge regarding the solution quality for the second objective value. Here, the ALNS cannot achieve the same performance as the solver. However, the average gap to the best solution reduces from 11 to 5% for more periods. This suggests that the heuristic is still valid for finding acceptable solutions for larger-scaled instances.

For the instances featuring 1880 periods, the average runtime is 47 min for HC and 24 min for RRT. HC finds better solutions in the first objective function (3% gap to RRT). RRT is more successful in the second objective but probably also the least restrictive because of the higher first objective value.

Figure 8 illustrates the progress of solution development when applying the ALNS based on HC or RTT to the large-scale instance. As visible in Fig. 8a, HC achieves the most improvement during the very first iterations. After the first hundred iterations, the first objective plateaus and only slowly improves. At the same time, the second objective incrementally decreases over time, interrupted by a temporary increment because of an improved first objective. Figure 8b shows that, for RRT, the first objective value oscillates between the current best and given threshold. As the threshold shrinks, the amplitude also shrinks. Overall, HC achieves the most improvement in the earlier iterations, whereas RRT might require even more iterations to achieve the same level. Running HC only for a low number of iterations, followed by a more intense search through RRT, might be an acceptable compromise for a low runtime and a good global solution quality.

Table 5 indicates the average number of times that each destroy and repair operator from Sect. 5 was applied in the large scenario. It also lists the average number of times the candidate solution was the new *best* solution, *better* than the current solution, *accepted* according to the chosen criterion, or *rejected*.

For HC, the candidate solution is either the new best, the same as the current solution (*accepted*), or worse than the current solution (*rejected*). *Random deletion and shift* is the most successful destroy operator, followed by the *passenger-driven* deletion. Nevertheless, HC regularly applies *freight-driven* deletion, and this accounts for approximately 13% of the iterations where the best solution is found. The *passenger-driven insertion* significantly outperforms the temporal-driven insertion. RRT stops after fewer iterations than HC. Here, the *passenger-driven* deletion induces more improvements than the *random deletion and shift*. A possible reason might be that the overall algorithm design includes more randomisation than HC and therefore, requires more operators for intensification.

Having thus validated and benchmarked the ALNS variants, we use HC to create schedules and allocate freight for larger instances that cannot be solved to optimality in an acceptable time. Based on the results, we analyse the system's sensitivity to capacity settings given different depot, freight-flow, and demand settings in the next section.

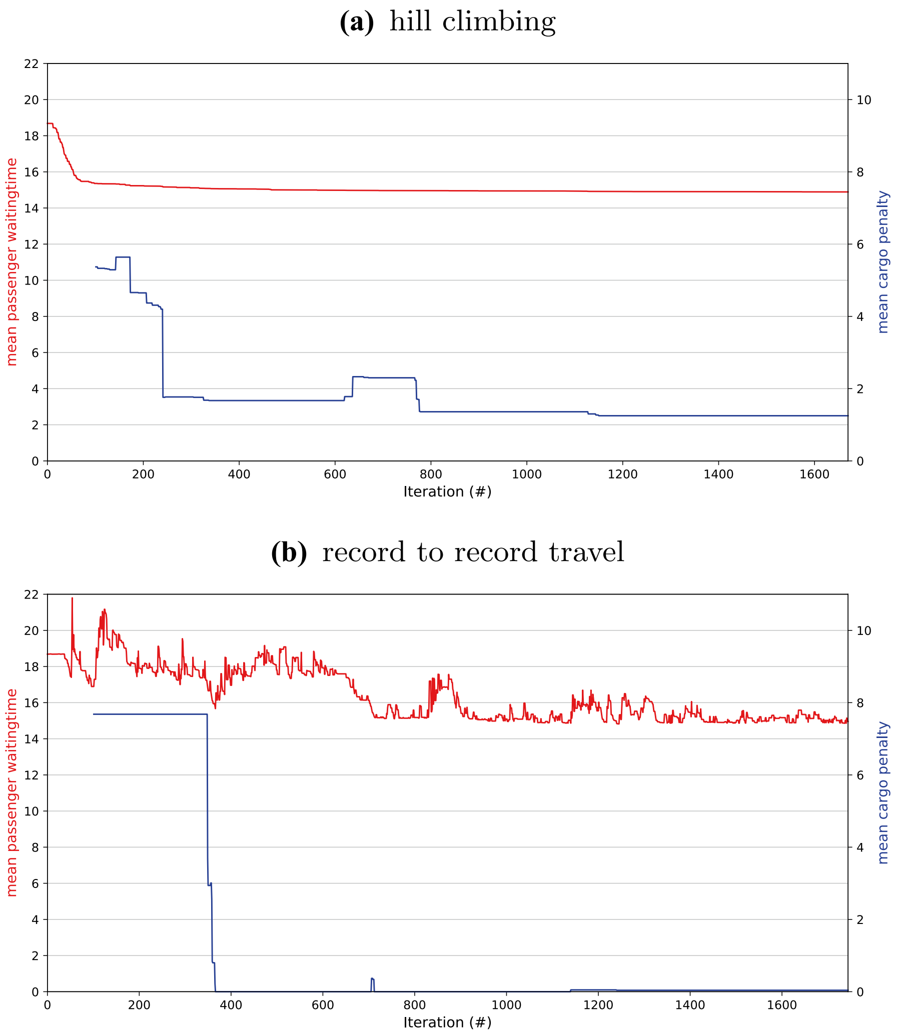


Fig. 8 Objective value across iterations for an exemplary instance

7 Sensitivity Analysis

When implementing a system for shared passenger and freight transport, the available capacities for freight transport, storage, and transhipment are important factors. While strategic fleet planning determines the vehicles' capacity, the strategic choice and design of micro-depots determine the capacity for storage and transhipment. To support such considerations, we showcase a sensitivity analysis that considers two depot-location settings and three freight-flow settings. Section 7.1 describes the experimental setting including the design of problem settings and instances. Subsequently, Sect. 7.2 presents the results for the computational experiments.

Table 5 Average count of operator usage for an exemplary set of instances

Crit	Operator	Total	Best	Better	Accepted	Rejected
RRT	Deletion and shift	450.8	14.4	198.7	224.9	12.8
RRT	Deletion (freight)	348.8	4.4	92.1	246.3	6
RRT	Deletion (passenger)	484.5	33.1	233	195.9	22.5
RRT	Insertion (passenger)	777.2	49.1	442	284.5	1.6
RRT	Insertion (time)	506.9	2.8	81.8	382.6	39.7
RRT	Total	1284.1	51.9	523.8	667.1	41.3
HC	Deletion and shift	638	109.8	-	184.7	343.5
HC	Deletion (freight)	376.6	23	-	5.7	347.9
HC	Deletion (passenger)	566.4	38.5	-	318.6	209.3
HC	Insertion (passenger)	1064.7	168.3	-	502	394.4
HC	Insertion (time)	516.3	3	-	7	506.3
HC	Total	1581	171.3	-	509	900.7

7.1 Instance and Scenario Design

In general, we consider a bi-directional network of ten equidistant stations, i.e., each service must pass twenty stops in total when going back and forth. The travel time between any two stops is 1 min. We limit the analysis to equidistant stations, as we also test for certain network characteristics.

Each experiment covers 1 day of operations. We discretise this time horizon into 2280 periods, such that each time period equals 30 s. Within the time horizon, given travel times, at most 100 services can circulate through the network.

To describe the temporal curve of the daily passenger demand, we extract the popular times for exemplary stations in the light rail network of Hamburg, Germany from [45]. According to a field study by [46], a single passenger needs 0.87 s on average to board or alight a vehicle. To account for simultaneous passenger processes in a vehicle with nine doors, we set the boarding and alighting parameters β^{board} and β^{alight} to $0.87/9 = 0.09$ s. In addition, we set a minimum dwell time $\beta^{m}in$ of 30 s to guarantee sufficient time for boarding and alighting processes. Furthermore, we expect a loading time of 20 s and an unloading time of 15 s per container. We assume that service times increase linearly with the number of passengers and containers. Each vehicle fits 500 passengers. The cargo releases are uniformly distributed between 7 am and 4 pm. The soft deadline is 1 h after release and the maximal delay is set to 1 h. The shared capacity for cargo ranges between 1 and 5 cargo containers for sensitivity analysis.

7.1.1 Freight Flow

To consider structural effects from differences in freight demand, we consider three freight-flow settings governing the probability of containers originating from certain arrival stops and ending at certain destination stops.

Uniform Freight Flow The first scenario considers *uniformly distributed freight demand*, i.e., each container may originate at or be destined for each stop with the same probability. We do not consider this a realistic setting but use it as a benchmark for the two alternative settings described in the following.

Central Freight In-flow The second setting implements the idea that goods are transhipped from long-distance transport in the suburbs for further transport to destinations in the city centre. Thus, most containers originate in the outer areas of the network and have destinations in the centre (cf. Fig. 9a).

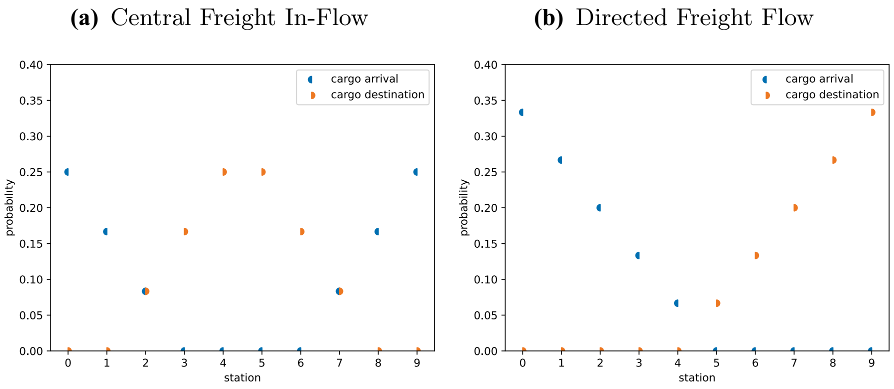
Directed Freight Flow The third scenario implements the idea that goods must be transported from an appropriate transhipment area in the city to an industrial area outside of the city. Hence, the majority of containers originate on one side of the network and must be transported to the other side (cf. Fig. 9b).

7.1.2 Depot Locations

Depending on the network layout, not all stops might be suitable for transhipping freight through micro-depots. As a strategic decision, the designers of a shared passenger and freight system must choose appropriate locations for depots. These choices result in alternative depot-location settings.

We assume that freight demand can arise at any stop, but that containers are always transported to the micro-depot closest to the original source and destination. These are the stops we term origin and destination stops here. To render different settings comparable, we adapt the deadline and release time of each container by adding up the travel time from the selected depot stops to the underlying source or destination.

For the sensitivity analysis, we consider two depot-location settings.


(a) Central Freight In-Flow

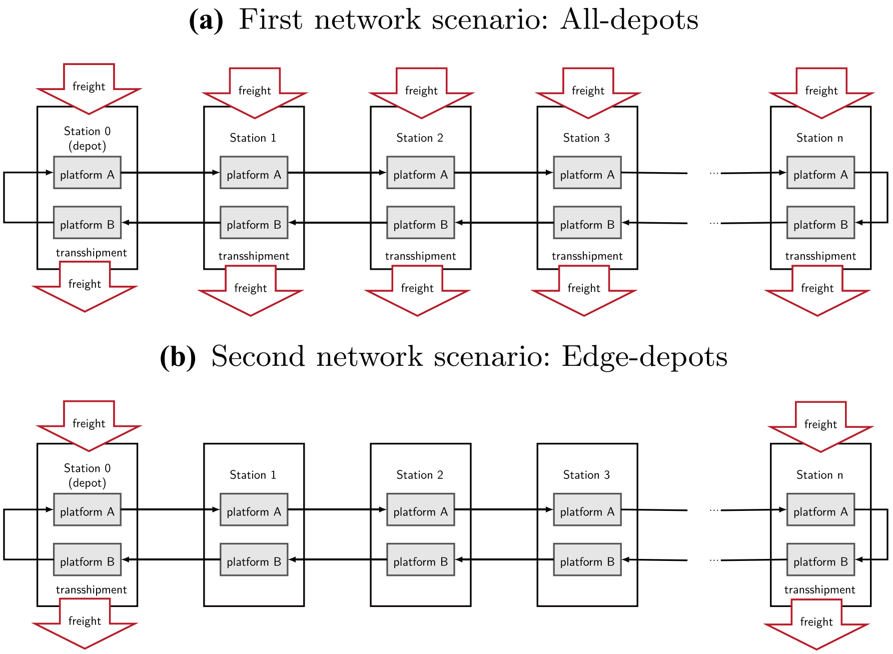
(b) Directed Freight Flow

Fig. 9 Probability distributions for the freight-flow-settings

All-Depots In this setting, all stops are available as micro-depots. Thus, all stops in the public transit networks offer sufficient capacity to store at least a small number of freight containers (cf. Fig. 10a). For the sensitivity analysis, the stop capacity ranges from one to five containers. To guarantee only minimal disturbance for the passenger service, we limit the maximal dwell time at each stop. All freight processes must be executable within a maximal dwell time of 1 min, i.e. two periods.

Edge-Depots The second setting assumes that the network does not allow for transhipment at intermediate stops. Instead, only stops that are situated at the outer edges of the network are depots (cf. Fig. 10b). To compensate for the lower depot density, we assume that the edge depots offer more capacity and consider stop capacities of two, four, six, eight, and ten containers in the sensitivity analysis. Since vehicles must also change directions at the related stations, additional dwell time results. We set the maximal dwell time in this design to 10 min, i.e., 20 periods.

To summarise, we consider three freight-flow settings, two depot-location settings, and twenty-five demand variants in the sensitivity analysis, creating 150 variations of the problem. As the freight demand is randomly drawn from a probability distribution, a single instance has only limited significance for the overall view of the problem. At the same time, we can only solve a limited number of instances, given a large number of problem variations. To balance efficiency and reliability, we randomly generate 10 instances for each of the 150 variations. Hence, we solve 1500 problem instances in total. All instances and result files are publicly available at [44].


(a) First network scenario: All-depots

(b) Second network scenario: Edge-depots

Fig. 10 Scenarios depending on the location and design of depots

7.2 Computational Results

We use the ALNS with hill climbing (HC) to solve the randomly generated instances generated from the parameter settings in the previous sub-section. As result indicators, we consider the passenger waiting time, the cargo acceptance rate, and the cargo delay. As expected, the passenger waiting time is stable across scenarios (results not shown). Because of the lexicographical objective, solutions with lower passenger waiting times are always prioritised. This results in only a 2% gap between the best and worst passenger waiting time found. Thus, the ALNS ensures a stable level of passenger service quality. From the maximal 100 services, all instances include at most five simultaneous services in peak times. Thus, the minimum fleet size to operate the computed time schedules is five vehicles.

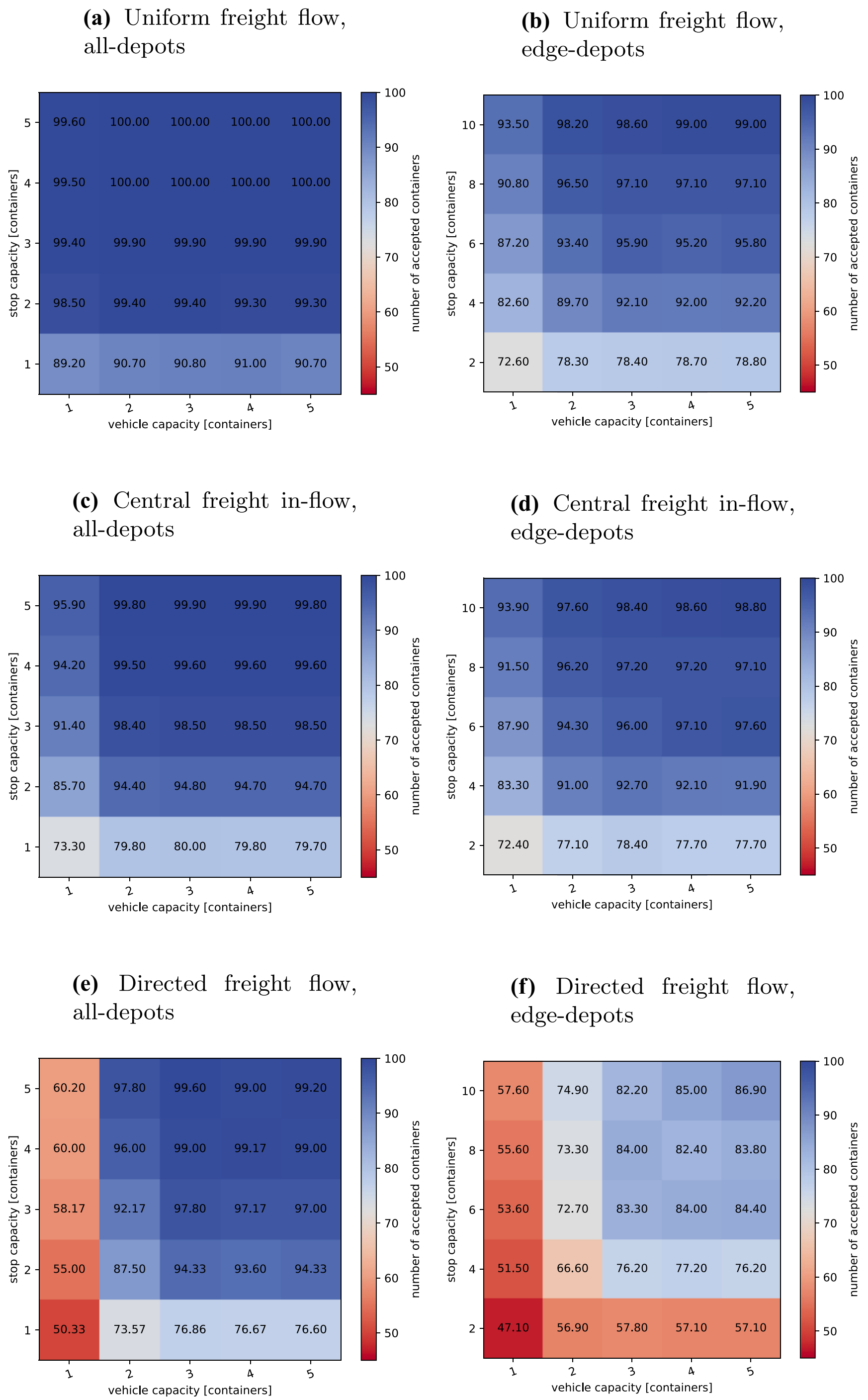
Figure 11 shows the average number of accepted cargo containers for varying vehicle and stop capacities. Following intuitive expectations, the all-depots setting is advantageous. The larger number of depots makes up for the more restrictive dwell times, letting this design outperform the turning-depot-design by 0.91 (for central freight flow and high capacity values) to 25.52% (for directed freight flow with minimal capacity values). To analyse under what settings the edge-depot setting enables good results, we further consider the structural characteristics.

First, we evaluate the results for the all-depot setting. When freight demand follows a uniform pattern, the average load per vehicle can be reduced to a minimum, and less vehicle and stop capacity is required. Figure 11a shows that the stop capacity threshold for accepting all freight is two and that this is independent of the vehicle capacity. If stops include only capacity for one container, the acceptance rate is still greater than 94%. Thus, the all-depot setting allows for an almost ideal acceptance rate given a uniform freight flow.

The central freight in-flow setting causes a lower acceptance rate (cf. Fig. 11c). As, in this setting, most freight originates at the outer stops, the required storage capacity at those stops increases. If a stop can only store a single container, the acceptance rate ranges between 73 and 80%, i.e., 11 to 18% less than for a uniform freight flow. The acceptance rate growth logarithmically with increasing stop-capacity. Containers with closely spaced release times will probably be picked up by the same service, leading to overlaps in storage time. The probability of overlapping storage times decreases with increasing capacity. The vehicle capacity also gains more relevance in the central freight in-flow setting. The acceptance rate rises by 4 to 18% when vehicles can carry an additional container. Given a uniform freight flow, certain containers are already delivered before new containers are picked up. This is much less likely under central freight in-flow.

For the directed freight-flow setting, freight originates at yet fewer depot stops. Additionally, the origin and destination stops do not overlap, i.e., all containers of the service must be picked up before any container reaches its destination. For this reason, the vehicle capacity becomes more crucial (see Fig. 11e). If there is only space for one cargo container in each vehicle, the acceptance rate ranges between 0 and 60%.

Next, we discuss the results for the edge-depots setting. As freight always originates from the nearest stop featuring a micro-depot, the origin distribution resulting



(a) Uniform freight flow, all-depots

(b) Uniform freight flow, edge-depots

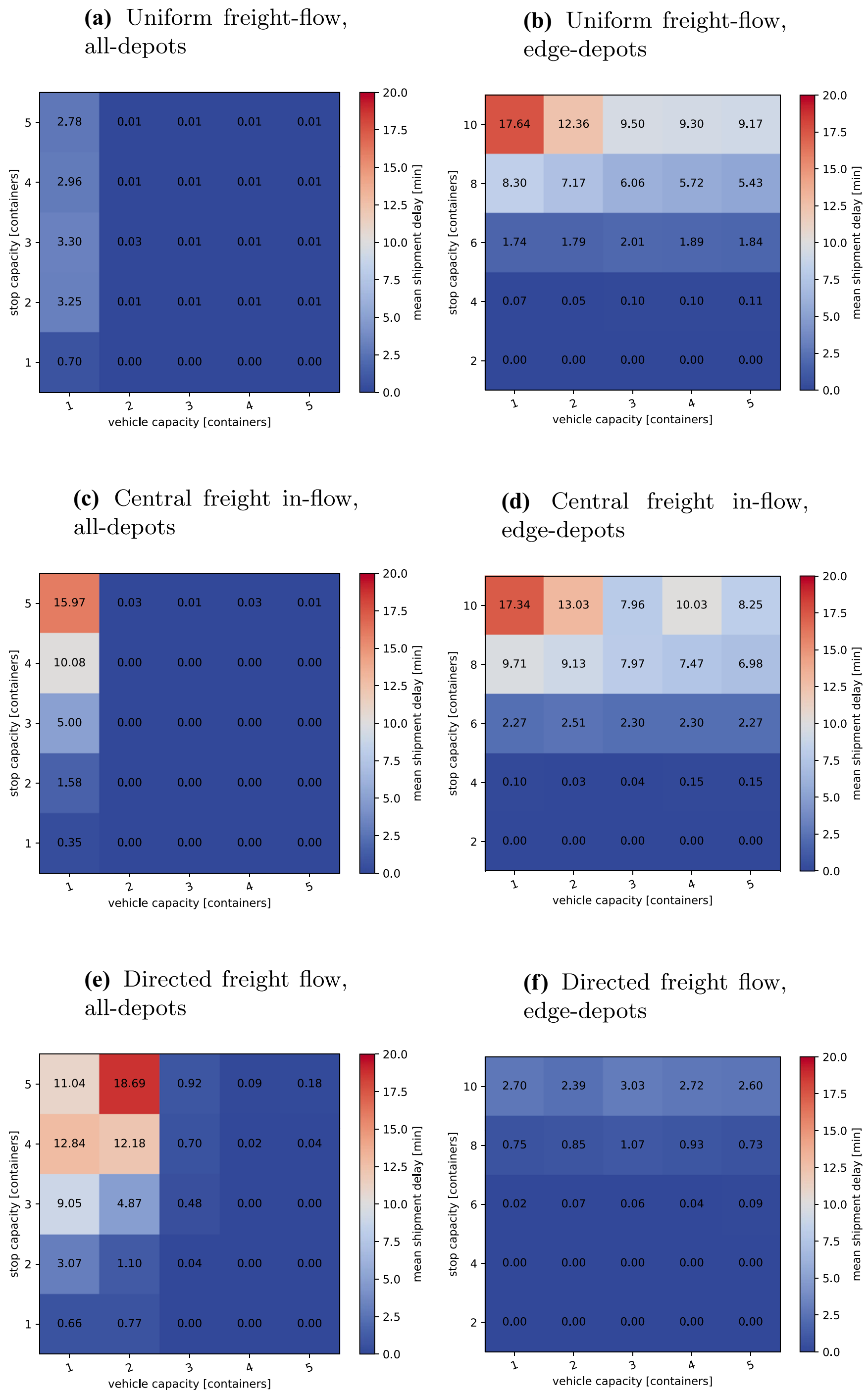
(c) Central freight in-flow, all-depots

(d) Central freight in-flow, edge-depots

(e) Directed freight flow, all-depots

(f) Directed freight flow, edge-depots

Fig. 11 Average number of accepted cargo containers for different scenarios


(a) Uniform freight-flow, all-depots

(b) Uniform freight-flow, edge-depots

(c) Central freight in-flow, all-depots

(d) Central freight in-flow, edge-depots

(e) Directed freight flow, all-depots

(f) Directed freight flow, edge-depots

Fig. 12 Average delay of accepted freight

from the uniform flow and the central freight-in-flow settings is the same given edge-depots. The results of the two depot-location settings are most similar for the central freight in-flow. From a practical perspective, a better use case for the edge-depot setting is the directed freight-flow setting. However, sufficient vehicle and stop capacity are crucial factors for the system's success (cf. Fig. 11b).

Figure 12 visualises the average delay of accepted containers in the various scenarios. Since we only evaluate the delay of accepted containers, a higher capacity does not automatically result in a lower delay. Overall, we observe low delay rates from 0 to 2 min. Especially with a high stop capacity and a low vehicle capacity, a high number of containers is accepted, but the average delay increases up to almost 19 min (cf. Figs. 11a, 12c, and e).

7.3 Managerial Implications

To highlight the managerial implications, we compare our findings to the computational study in [10]. In particular, this predecessor model differed from the one considered here in that it assumed both unlimited stop capacity and that all freight originates from a single depot.

In general, the findings of [10] emphasise the importance of optimisation techniques to satisfy passenger and freight demand simultaneously. In that setting, a shared capacity between passengers and freight appeared more suitable for urgent containers with a tight deadline than a system with vehicles dedicated to either passenger or freight transportation. Furthermore, [10] highlight the impact of the freight loading and unloading processes. In that study, the dwell time represents the bottleneck in the considered system and the vehicle capacity plays only a minor role.

Beyond demonstrating that the ALNS is an appropriate alternative to solving the problem to optimality, our results indicate that the storage capacity of micro-depots is an essential factor for the system's success when allowing for pick-ups beyond a central depot. Additionally, the dwell time still has a high impact on the system's service quality. Especially for the edge-depot network design, a sufficient maximal dwell time is crucial to enable acceptable acceptance rates. However, for a high micro-depot capacity but a low vehicle capacity, the solutions can accept most of the containers but lack punctuality. Thus, we conclude that the shared capacity mode is only suited for urgent parcel containers when both stops and vehicles offer sufficient freight capacity.

8 Conclusion

In this work, we considered the problem of scheduling services and allocating cargo in a shared passenger and freight transit system featuring multiple micro-depots with limited storage capacity. To prioritise the passenger service, we considered a lexicographic objective function, where the first of the two objectives is to increase the average passenger waiting time across stops. The second objective function reduces the average freight delay. Furthermore, we presented an adaptive large neighbourhood search (ALNS) to find good solutions for large-scale problem instances. As acceptance criteria, we implemented and compared hill climbing and record-to-record travel.

The performance analysis results indicated that the ALNS is appropriate for solving realistic-sized problems. The hill climbing approach achieved the best improvement in early iterations, while the record-to-record travel required more iterations to adapt.

Sharing resources, e.g., vehicle capacity, is one of the main advantages of such a system and one of its main challenges. We considered the effect of freight flow settings and depot locations in a sensitivity analysis for varying vehicle and stop capacities. Due to the lexicographic objective function, we found that the average waiting time was stable for various settings. The result showed that dense freight demand flows require sufficient depot and vehicle capacity.

Lower storage capacities can suffice if every stop is a micro-depot. However, this requires sufficient space at each stop to handle the transhipment process and to store containers. Additionally, the design might come with higher initial costs because of the reconstruction work to enable transhipping at depot stops. At the same time, the realism of an edge-depot setting under a central freight flow is questionable. Apart from the adaptation of release and deadline times, the distance between the depot station and desired demand locations might be too large to ensure a valid satisfactory transportation service.

Our results when evaluating freight delay illustrate that a high stop capacity is insufficient to reach a high service degree if the cargo delay is also an important factor. A high storage capacity at the stops may ensure a high acceptance rate for freight, but vehicle capacity is an important factor in reducing delays.

The model presented here only considers a single line. Extending the model to a network of intersecting lines, where freight can change lines at transhipment depots, is an interesting concept for future work. Further extensions could also include the problem of rolling stock, i.e., assigning the vehicles in the fleet to services instead of setting an approximated upper boundary for the number of services. In the computational study, we evaluated a network with closely situated stops for each station. In future work, a transit system with sparsely distributed stops is also worth analysing. Additionally, this work did not consider further transport from the micro-depots over the "last meter". Future work could also anticipate the capacity request acceptance over time to avoid late denials. At the moment, the ALNS algorithm solves the freight assignment problem with a commercial solver. The evaluation of different algorithms to find a feasible assignment and further accelerate the heuristic might also be a promising approach.

Author Contribution LH created the mathematical model, implemented the code, ran the experiments, prepared the figures, and wrote the methodological parts of the manuscript. CC supervised and advised on the research and modelling and wrote the introduction. Both authors contributed to writing the literature review and the conclusion and reviewed and edited the manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL.

Availability of Data and Material Data are publicly available at https://doi.org/10.5281/zenodo.7576384.

Code Availability Code is publicly available at https://doi.org/10.5281/zenodo.7576384.

Declarations

Ethics Approval Not applicable

Consent to Participate Not applicable

Consent for Publication Not applicable

Conflict of Interest The authors declare no competing interests.

References

1. Kiba-Janiak M, Marcinkowski J, Jagoda A, Skowrońska A (2021) Sustainable last mile delivery on e-commerce market in cities from the perspective of various stakeholders. Literature review. Sustain Cities Soc 71:102984. https://doi.org/10.1016/j.scs.2021.102984
2. Bosse A, Ulmer MW, Manni E, Mattfeld DC (2023) Dynamic priority rules for combining on-demand passenger transportation and transportation of goods. Eur J Oper Res 309(1):399–408. https://doi.org/10.1016/j.ejor.2023.01.010
3. Ghilas V, Cordeau J-F, Demir E, Woensel TV (2018) Branch-and-price for the pickup and delivery problem with time windows and scheduled lines. Transp Sci 52(5):1191–1210. https://doi.org/10.1287/trsc.2017.0798
4. De Langhe K, Meersman H, Sys C, Van de Voorde E, Vanelslander T (2019) How to make urban freight transport by tram successful? J Ship Trade 4(1):13. https://doi.org/10.1186/s41072-019-0055-4
5. Frey M (2023) Website of the LogIKTram project. https://logiktram.de/en. Accessed 28 Mar 2023
6. Arvidsson N, Browne M (2013) A review of the success and failure of tram systems to carry urban freight: the implications for a low emission intermodal solution using electric vehicles on trams, vol 54. Technical Report No. 5, European Transport / Trasporti Europei
7. Cavallaro F, Nocera S (2021) Integration of passenger and freight transport: a concept-centric literature review. Res Transp Bus Manag 43:100718. https://doi.org/10.1016/j.rtbm.2021.100718
8. Delle Donne D, Alfandari L, Archetti C, Ljubić I (2023) Freight-on-transit for urban last-mile deliveries: a strategic planning approach. Transp Res B: Methodol 169:53–81. https://doi.org/10.1016/j.trb.2023.01.004
9. Li Z, Shalaby A, Roorda MJ, Mao B (2021) Urban rail service design for collaborative passenger and freight transport. Transp Res E: Logist Transp Rev 147:102205. https://doi.org/10.1016/j.tre.2020.102205
10. Hörsting L, Cleophas C (2023b) Scheduling shared passenger and freight transport on a fixed infrastructure. Eur J Oper Res 306(3):1158–1169. https://doi.org/10.1016/j.ejor.2022.07.043
11. Yin J, D'Ariano A, Wang Y, Yang L, Tang T (2021) Timetable coordination in a rail transit network with time-dependent passenger demand. Eur J Oper Res 295(1):183–202. https://doi.org/10.1016/j.ejor.2021.02.059
12. Cochrane K, Saxe S, Roorda MJ, Shalaby A (2017) Moving freight on public transit: best practices, challenges, and opportunities. Int J Sustain Transp 11(2):120–132. https://doi.org/10.1080/15568318.2016.1197349
13. Azcuy I, Agatz N, Giesen R (2021) Designing integrated urban delivery systems using public transport. Transp Res E: Logist Transp Rev 156:102525. https://doi.org/10.1016/j.tre.2021.102525

14. Motraghi A, Marinov MV (2012) Analysis of urban freight by rail using event based simulation. Simul Model Pract Theory 25:73–89. https://doi.org/10.1016/j.simpat.2012.02.009
15. van Heeswijk WJA, Mes MRK, Schutten JMJ, Zijm WHM (2020) Evaluating urban logistics schemes using agent-based simulation. Transp Sci 54(3):651–675. https://doi.org/10.1287/trsc.2019.0971
16. Katsela K, Güneş Ş, Fried T, Goodchild A, Browne M (2022) Defining urban freight microhubs: a case study analysis. Sustainability 14(1):532. https://doi.org/10.3390/su14010532
17. Zhao L, Li H, Li M, Sun Y, Hu Q, Mao S, Li J, Xue J (2018) Location selection of intra-city distribution hubs in the metro-integrated logistics system. Tunn Undergr Space Technol 80:246–256. https://doi.org/10.1016/j.tust.2018.06.024
18. Assmann T, Lang S, Müller F, Schenk M (2020) Impact assessment model for the implementation of cargo bike transshipment points in urban districts. Sustainability 12(10):4082. https://doi.org/10.3390/su12104082
19. Alumur SA, Campbell JF, Contreras I, Kara BY, Marianov V, O'Kelly ME (2021) Perspectives on modeling hub location problems. Eur J Oper Res 291(1):1–17. https://doi.org/10.1016/j.ejor.2020.09.039
20. Gouveia L, Leitner M, Ruthmair M (2022) Multi-depot routing with split deliveries: models and a branch-and-cut algorithm. Transportation Science. Ahead of print. https://doi.org/10.1287/trsc.2022.1179
21. Brandão J (2020) A memory-based iterated local search algorithm for the multi-depot open vehicle routing problem. Eur J Oper Res 284(2):559–571. https://doi.org/10.1016/j.ejor.2020.01.008
22. Ramos TRP, Gomes MI, Barbosa-Póvoa AP (2020) A new matheuristic approach for the multi-depot vehicle routing problem with inter-depot routes. OR Spectr 42(1):75–110. https://doi.org/10.1007/s00291-019-00568-7
23. Montoya-Torres JR, López Franco J, Nieto Isaza S, Felizzola Jiménez H, Herazo-Padilla N (2015) A literature review on the vehicle routing problem with multiple depots. Comput Ind Eng 79:115–129. https://doi.org/10.1016/j.cie.2014.10.029
24. Mousavi K, Bodur M, Roorda MJ (2022) Stochastic last-mile delivery with crowd-shipping and mobile depots. Transp Sci 56(3):612–630. https://doi.org/10.1287/trsc.2021.1088
25. Zhen L, Baldacci R, Tan Z, Wang S, Lyu J (2022) Scheduling heterogeneous delivery tasks on a mixed logistics platform. Eur J Oper Res 298(2):680–698. https://doi.org/10.1016/j.ejor.2021.06.057
26. Hof J, Schneider M (2021) Intraroute resource replenishment with mobile depots. Transp Sci 55(3):660–686. https://doi.org/10.1287/trsc.2020.1034
27. Kliewer N, Mellouli T, Suhl L (2006) A time–space network based exact optimization model for multi-depot bus scheduling. Eur J Oper Res 175(3):1616–1627. https://doi.org/10.1016/j.ejor.2005.02.030
28. Amberg B, Amberg B (2023) Robust and cost-efficient integrated multiple depot vehicle and crew scheduling with controlled trip shifting. Transp Sci 57(1):82–105. https://doi.org/10.1287/trsc.2022.1154
29. Røpke S, Pisinger D (2006) An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transp Sci 40(4):455–472. https://doi.org/10.1287/trsc.1050.0135
30. Pisinger D, Røpke S (2010) Large neighborhood search. In: Gendreau M (ed) Handbook of Metaheuristics, 2nd edn. Springer, pp 399–419
31. Masson R, Lehuédé F, Péton O (2013) An adaptive large neighborhood search for the pickup and delivery problem with transfers. Transp Sci 47(3):344–355. https://doi.org/10.1016/j.cor.2020.105110
32. Ghilas V, Demir E, Van Woensel T (2016) An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows and scheduled lines. Comput Oper Res 72:12–30. https://doi.org/10.1016/j.cor.2016.01.018
33. Hemmelmayr VC, Cordeau J-F, Crainic TG (2012) An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. Comput Oper Res 39(12):3215–3228. https://doi.org/10.1016/j.cor.2012.04.007
34. Enthoven DLJU, Jargalsaikhan B, Roodbergen KJ, uit het Broek MAJ, Schrotenboer AH (2020) The two-echelon vehicle routing problem with covering options: city logistics with cargo bikes and parcel lockers. Comput Oper Res 118:104919. https://doi.org/10.1016/j.cor.2020.104919
35. Bilgin B, De Causmaecker P, Rossie B, Vanden Berghe G (2012) Local search neighbourhoods for dealing with a novel nurse rostering model. Ann Oper Res 194(1):33–57. https://doi.org/10.1007/s10479-010-0804-0
36. Kovacs AA, Parragh SN, Doerner KF, Hartl RF (2012) Adaptive large neighborhood search for service technician routing and scheduling problems. J Sched 15(5):579–600. https://doi.org/10.1007/s10951-011-0246-9
37. Cinar A, Salman FS, Bozkaya B (2021) Prioritized single nurse routing and scheduling for home healthcare services. Eur J Oper Res 289(3):867–878. https://doi.org/10.1016/j.ejor.2019.07.009

38. Windras Mara ST, Norcahyo R, Jodiawan P, Lusiantoro L, Rifai AP (2022) A survey of adaptive large neighborhood search algorithms and applications. Comput Oper Res 146. https://doi.org/10.1016/j.cor.2022.105903
39. Barrena E, Canca D, Coelho LC, Laporte G (2014b) Single-line rail rapid transit timetabling under dynamic passenger demand. Transp Res B: Methodol 70:134–150. https://doi.org/10.1016/j.trb.2014.08.013
40. Barrena E, Canca D, Coelho LC, Laporte G (2014a) Exact formulations and algorithm for the train timetabling problem with dynamic demand. Comput Oper Res 44:66–74. https://doi.org/10.1016/j.cor.2013.11.003
41. Yuan J, Gao Y, Li S, Liu P, Yang L (2022) Integrated optimization of train timetable, rolling stock assignment and short-turning strategy for a metro line. Eur J Oper Res 301(3):855–874. https://doi.org/10.1016/j.ejor.2021.11.019
42. Schöbel A (2012) Line planning in public transportation: models and methods. OR Spectr 34(3):491–510
43. Santini A, Røpke S, Hvattum LM (2018) A comparison of acceptance criteria for the adaptive large neighbourhood search metaheuristic. J Heuristics 24(5):783–815. https://doi.org/10.1007/s10732-018-9377-x
44. Hörsting L, Cleophas C (2023a) Integrating micro-depot freight transport in existing public transport services. Zenodo. Zenodo. https://doi.org/10.5281/zenodo.7576384 ([Code and data])
45. Google (2021) Popular times, wait times, and visit duration. Google Help Center. https://support.google.com/business/answer/6263531?hl=en. Online. Accessed 01 Dec 2021
46. Li Z, Lo S, Ma J, Luo X (2020) A study on passengers' alighting and boarding process at metro platform by computer simulation. Transp Res A Policy Pract 132:840–854. https://doi.org/10.1016/j.tra.2019.12.017