ARTICLE IN PRESS

European Journal of Operational Research xxx (xxxx) xxx

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Production, Manufacturing, Transportation and Logistics

The location routing problem with time windows and load-dependent travel times for cargo bikes

Alexander Rave [*], Pirmin Fontaine

Catholic University Eichstätt-Ingolstadt, Ingolstadt School of Management & Mathematical Institute for Machine Learning and Data Science, Auf der Schanz 49, 85049 Ingolstadt, Germany

ARTICLE INFO

Keywords:
Routing
Last-mile delivery
MILP
ALNS

ABSTRACT

Last-mile delivery with traditional delivery trucks is ecologically unfriendly and leads to high road utilization. Thus, cities seek for different delivery options to solve these problems. One promising option is the use of cargo bikes in last-mile delivery. These bikes are typically released at micro hubs, which are small containers or facilities located at advantageous places in the city center. Since the bike's travel speed depends on its remaining load and the street gradient, placing the hubs at valleys might cause additional work for rides. Therefore, the following question arises: How high is the impact of load-dependent travel times on micro hubs' cost-optimal placements?

To answer this question, we introduce the location routing problem with time windows and load-dependent travel times. We formulate the problem as a mixed-integer linear program and introduce an adaptive large neighborhood search with a problem-specific procedure for micro hub placements and problem-specific operators to solve larger instances. In a numerical study, we find that load-dependent travel times significantly influence the location of hubs, following that hubs with a higher elevation are preferably used. Moreover, customers are served from hubs with a similar elevation. This would not be the case if load-dependent travel times are ignored, resulting in an increase in costs by up to 2.7% or, instead, to up to 26% infeasible solutions as time windows are not adhered to.

1. Introduction

Cities are increasingly struggling with higher road utilization, leading to traffic jams and environmental pollution. Last-mile parcel transport, in particular, intensifies these problems, as delivery vehicles, typically vans or trucks, are often parked in the second row. Thus, cities and also parcel service providers are looking for alternative concepts, as traffic jams and parking problems reduce profit margins and delay deliveries (Boysen et al., 2021). Parcel delivery via cargo bikes is a promising option in last-mile delivery because they can ride on the bicycle path and are therefore independent of road traffic. Thus, DHL uses cargo bikes for last-mile delivery, for example, in Miami (DHL, 2020) or Edinburgh (DHL, 2021).

One major disadvantage of these bikes is that they have a lower average speed that is especially dependent on their load and the gradient of the street (Fontaine, 2022) and a shorter range compared to traditional delivery trucks. To overcome these disadvantages, bikes can be released at micro hubs, which are small containers that can be placed at multiple places within a city (e.g., Berger et al., 2007). These micro hubs (in combination with cargo bikes) are, for example, used by Amazon in Berlin Amazon (2024) or by UPS in Hamburg (Logistra, 2022).

Considering the peculiarities of cargo bikes, i.e., the load-dependent travel times, locating hubs at valleys might be unfavorable as additional work arises for rides, making bikes travel at lower speeds. These longer tours delay customer service, which is especially problematic if customers need to be served within specific time periods, i.e., time windows. Contrary, considering time windows and consolidation effects, choosing the hubs with the highest elevation is not an all-purpose solution. Thus, questions arise on where to place these micro hubs and how high load-dependent travel times impact micro hubs' placements.

To answer these questions, it is necessary to consider the operational routing and tactical micro hub location decisions simultaneously, which is in the literature referred to as the location routing problem (LRP) (e.g., Berger et al., 2007). We extend this LRP by time windows and the peculiarities of cargo bikes, namely the load-dependent travel times, and introduce it as a location routing problem with time windows and load-dependent travel times (LRPTWLTT). To tackle this problem, we formulate it as a mixed-integer linear program (MILP)

* Corresponding author.
E-mail addresses: arave@ku.de (A. Rave), pirmin.fontaine@ku.de (P. Fontaine).

https://doi.org/10.1016/j.ejor.2024.11.040
Received 3 July 2024; Accepted 25 November 2024
Available online 3 December 2024

and introduce an adaptive large neighborhood search (ALNS) that separates the tactical location and fleet decisions from the operators by including a problem-specific procedure for micro hub placements. As a result, only four problem-specific operators are required to tackle the operational routing, making it easier to implement compared to comparable ALNS for the LRP or LRP with time windows (LRPTW) (e.g., Hemmelmayr et al., 2012).

We contribute to the literature in four ways: First, we combine an LRPTW with load-dependent travel times, or in other words, we extend the operational routing problem with load-dependent travel times of Fontaine (2022) by the tactical micro hub location decision. Second, we present a formal description of the LRPTWLTT as MILP. Third, we develop an ALNS that separates the tactical decision from the operators, following that only a few operators are required. Fourth, we generate multiple managerial insights on both the impact of load-dependent travel times and the location decision.

The structure of this paper is as follows: in Section 2, we describe the problem setting in detail. In Section 3, we introduce the most relevant literature and delimit this paper from the literature. We formulate the problem as MILP in Section 4 and present the ALNS in Section 5. In Section 6, we show the numerical results. Last, Section 7 summarizes our findings and gives a brief outlook.

2. Problem setting

We decide on the cost-minimal locations of micro hubs, the number of (homogeneous) cargo bikes allocated to each hub, and the bikes' routing to serve all customers, who have their individual demand mass, once. For this, we consider fixed costs for each micro hub and cargo bike used and variable costs for load-dependent traveling times, which is a special case of asymmetric routing costs. The variable costs are only dependent on the traveling times as energy costs are marginal if there are any, and the bike costs are already excluded as a fixed term. Thus, the driver's salary is the main cost driver (variable costs). We further show how to adjust our objective to symmetric routing costs that are dependent on the traveled distance.

We assume a finite set of potential micro hub locations and (opened) hubs are fully loaded with parcels at the beginning of the considered delivery time period. This can be practically ensured by supplying all hubs before the first working shift starts, e.g., early in the morning. Please note that we do not consider the routing to supply these micro hubs as the geographical area under consideration is limited due to the low speed and, therefore, the cargo bikes' range. In the considered instances in our case study, the maximum distance between two nodes, if they are at opposite corners of the area, lies between 2.3 km and 4.7 km, dependent on the instance. If a large city is to be supplied by bikes, it can be divided into different zones, e.g., postcodes. Additional different costs incurred by the hubs' supply could also be reflected in the fixed costs of the hubs.

Multiple cargo bikes can be allocated to a single hub with both hubs and bikes having a certain weight-dependent capacity. Thus, once allocated to a micro hub, the cargo bike starts and ends its route at this hub. Note that the number of bikes allocated to a hub is not restricted (e.g., Ponboon et al., 2016). This is because the bikes are not parked in the hubs themselves. They are parked next to them, or the delivery staff will drive up with them. Please note that considering a hub capacity of 600 kg and a bike capacity of 150 kg, as in our case study, there are not necessarily up to four bikes per hub. Especially due to the time windows, it might be reasonable to have a hub with, e.g., eight half-loaded bikes. However, there will not be an arbitrary number of bikes used due to their associated fixed costs.

We consider one representative demand, i.e., one single demand scenario that reflects the demand of the considered area well. This is a typical assumption in tactical or strategic planning (e.g., Canca et al., 2019; Rave et al., 2023a). Demand variations are then adhered to by operational day-ahead planning for routing and allocating bikes to hubs. In Section 6.3.2, we show that considering such a single demand scenario instead of multiple demand scenarios leads to a high-quality solution.

There are load-dependent travel times that influence the arrival times at each customer. The load-dependent travel times depend on the air and rolling resistance, the gravity force, and frictional losses, which are influenced by the transported mass (Fontaine, 2022). Each customer has a fixed time window where he can be served. The bike might arrive earlier at the customer, which leads to waiting times, but no extra costs. On the contrary, it is strictly prohibited that the bike arrives later, i.e., the time windows are hard (Ponboon et al., 2016). In addition, service times arise when a customer is served.

3. Literature review

This section presents the most relevant literature on the location routing problem without and with time windows and routing problems with load dependency. For an extensive review of location routing problems, we recommend the literature review of Drexl and Schneider (2015).

3.1. Location routing problem

Salhi and Rand (1989) are one of the first that combine the location and routing decision and find that it is worth considering both simultaneously. The LRP is then extended by multiple publications, e.g., by Berger et al. (2007), who introduce a MILP formulation and a Branch-and-Price Algorithm for an LRP with a limitation on the maximum length of a route.

Heuristic solution methods are commonly used to solve larger instances. So, Tuzun and Burke (1999) introduce a two-phase tabu search for an LRP that has uncapacitated hubs and Barreto et al. (2007) derive a cluster analysis based heuristic for an LRP with capacitated hubs. Hemmelmayr et al. (2012) introduce an ALNS for an LRP and a two-echelon vehicle routing problem (2E-VRP). Voigt et al. (2022) extend the typical ALNS structure for the LRP, the 2E-VRP, and the multi-depot vehicle routing problem (MDVRP) by deriving a hybrid ALNS, which combines a genetic algorithm with an ALNS.

3.2. Location routing problem with time windows

LRPTWs are hardly represented in the literature, also found by Drexl and Schneider (2015). So, Ponboon et al. (2016) develop a branch-and-price solution approach for an LRPTW and present optimally solved benchmark instances. To solve larger LRPTW instances, Maghfiroh et al. (2023) present a variable neighborhood search with a path relinking algorithm.

Schiffer and Walther (2017) focus on an LRPTW when considering electric cars with a limited driving range, longer charging times, and partial recharging. Schiffer and Walther (2018) extend this problem setting by additionally considering intra-route facilities, e.g., charging stations that must be visited to keep the vehicle operational. The authors present a problem-specific ALNS that includes a local search and a dynamic programming component that is executed in an iteration if a promising solution is found.

3.3. Routing problems with load-dependency

Bektaş and Laporte (2011) initially consider load-dependency in the objective function and introduce the pollution routing problem (PRP), a variant of the VRP where emissions are minimized. In contrast to our load-dependent travel times, the load-dependency is only in the objective and does not influence the arrival times at customers and their time windows. To solve larger instances, Demir et al. (2012) develop an ALNS for the PRP. Kramer et al. (2015) also consider the PRP and present a matheuristics combining integer programming with a local search and a speed optimization algorithm to find each arc's optimal speeds.

Considering load-dependency both in the objective and in the arrival times at customers, Fontaine (2022) introduces the VRP with load-dependent travel times and time windows for cargo bikes. Mühlbauer and Fontaine (2021) also consider cargo bikes, but no load-dependent travel times and asymmetric distances instead. Our load-dependent travel times extend this, as our asymmetry depends on the vehicle's load and the gradient of the street. Asymmetry is also considered by uit het Broek et al. (2021), who present a branch-and-cut algorithm for routing problems, including an LRP. The authors consider asymmetry in the cost structure.

Load-dependency is also considered in the drone literature. So, drones' load-dependent speeds and, thus, travel times are considered by Nishira et al. (2023), who extend the truck-drone tandem of Murray and Chu (2015). The authors approximate the non-linear speed function by a linear and quadratic regression. Tamke and Buscher (2023) and Dukkanci et al. (2021) consider a truck-drone variant, where drone speeds are not the result of their loads, but the speed is a decision influencing the drones' energy consumption. Rave et al. (2023b) also consider different loads in drone tours, but this influences only the costs of a flight.

Considering drones, the literature typically refers to the (non-linear) load-dependent energy consumption. This means that larger payloads reduce the drone's flight duration. The drone's speed, however, is not affected by the payload. So, Jeong et al. (2019) consider this load-dependent energy consumption in a single truck, single drone routing problem, Xia et al. (2023) for multiple trucks and drones, Cheng et al. (2020) in a routing problem with multiple drones (no trucks), and Bruni et al. (2023a, 2023b) in an LRP with drones. Bruni et al. (2023a) additionally take flight time uncertainty for drones into account. Considering electric-powered vehicles in general, e.g., the electric VRP, load-dependent energy consumption is found to be relevant for the recharging times but not assumed to influence travel times (Kancharla & Ramadurai, 2020; Wu et al., 2023).

3.4. Further relevant routing problems

Further comparable problem settings are the truck and trailer problem (TTRP) and the line-haul feeder VRP (FVRP), which are extensions of the VRP. In the TTRP, a truck has the possibility to carry a trailer. As the trailer cannot be taken to all customers, it must be parked at certain nodes before the truck continues its tour. The trailer must be picked up at the same node (Lin et al., 2009). Rothenbächer et al. (2018) extend this problem by time windows and introduce a variant with quantity-dependent time for transferring load from truck to trailer.

In the FVRP, two types of customers are served by two types of vehicles, including potential transshipments between both vehicles. Formally introduced by Brandstätter and Reimann (2018), it has also been considered by Huang et al. (2019). Sarbijan and Behnamian (2022) extended it to a dynamic version and added flexible time windows.

3.5. Summary

On the one hand, the LRP literature does not consider load-dependent travel times with Bruni et al. (2023a) and Bruni et al. (2023b) being closest considering an LRP with load-dependent energy consumption. On the other hand, the tactical location and fleet allocation decisions are missing when considering load-dependent travel times as in Fontaine (2022). Our paper fills this gap and connects both literature streams.

4. Mathematical model

In this section, we first show how to model load-dependent travel times, and second, we introduce our MILP formulation.

4.1. Modeling load-dependent traveling times

When traveling from node i to node j, traveling times depend on the remaining bike's load, the rolling resistance F^r, the gravity force F^g, the air resistance F^d, and the available power P of the cyclist and the bike's battery. Moreover, F^r and F^g are significantly influenced by the road's gradient. To account for the non-linearity of the load-dependent travel times, we follow the idea of Fontaine (2022) and Bektaş and Laporte (2011), and divide the bike's load into several load levels defined by set L. This allows the different loads to be converted into a piecewise linear function. Each load level l is bounded by its minimum p_l and the maximum r_l and assumed to have an average load of $(p_l + r_l)/2$. Then, p_0 equals 0 and $p_{|L|}$ equals Q^b, with Q^b being the maximum payload per bike. In Section 6.3.1, we show that the linear approximation does not lead to a loss of precision when the number of load levels is sufficiently large, which is the case for even ten load levels in our numerical study.

Algorithm 1 shows the pseudo-code for calculating the load-dependent travel times $(t_{i,j,l})$ in a pre-processing step as in Fontaine (2022). For each load level $l \in L$ and nodes $i, j \in N$, the required power P is calculated, which is based on F^r, F^g, and F^d. F^r and F^g are based on the load m_l and the slope $h_{i,j}$, which represents the road gradient in rad when traveling directly between i and j. m_l includes the weight of the bike and the cyclist as well as the payload defined by the load-interval $[p_l, r_l]$. As long as the required power P to travel at a certain speed v exceeds the available power $Power$, v and, thus, F^d and P are reduced. Last, the load-dependent traveling time is computed with the resulting speed v.

Algorithm 1: Algorithm for determining load-dependent travel times

for $l \in L, i, j \in N$ **do**
  $F^r = C_r \cdot g \cdot m_l \cdot \cos(\arctan(h_{i,j}))$
  $F^g = g \cdot m_l \cdot \sin(\arctan(h_{i,j}))$
  $F^d = \frac{C_d \cdot \rho \cdot A \cdot v^2}{3.6^2 \cdot 2}$
  $P = (F^r + F^g + F^d) \cdot \frac{v}{3.6 \cdot 0.95}$
  while $P > Power$ **do**
    $v \leftarrow v - 0.01$
    $F^d \leftarrow \text{Update}(F^d)$
    $P \leftarrow \text{Update}(P)$
  end
  $t_{i,j,l} = \frac{d_{i,j}}{v}$
end

Further required are the following parameters (Bombach, 2016; Fontaine, 2022; Wilson & Schmidt, 2020): The initial speed $v = 25$ km/h, the gravity force $g = 9.81$ m/s^2, air density $\rho = 1.18$ kg/m^3, the cross-sectional front area of the bicycle, the cargo, and the rider $A = 0.83$ m^2, the coefficient of drag $C_d = 1.18$, the rolling resistance $C_r = 0.01$.

4.2. MILP formulation

We consider the index set for customers (C), micro hubs (H), and all nodes $(N = H \cup C)$. Each customer $c \in C$ has a time window $[a_c, b_c]$, a demand mass q_c, and a service time s_c. Each bike has a capacity of Q^b and each hub of Q^h. In the delivery system, costs occur for each hub opened $(\hat{c}^{f,h})$, each bike used $(\hat{c}^{f,b})$, and per time unit traveling (c^v).

The main decision variables determine the routing between nodes $(x_{i,j} \in \{0,1\})$ if a micro hub is opened $(y_i \in \{0,1\})$, and the number of bikes allocated to a hub $(w_i \in \mathbb{N})$. Aligned decisions are the arrival time at a node $(v_i \in \mathbb{R}^+)$. The decision variable $z_{i,j,l} \in \{0,1\}$ indicates whether a load level is chosen when traveling from i to j. Further,

Table 1
List of notation of the MILP.

Index sets, parameters and variables	
H	Index set for micro hubs
C	Index set for customers
N	Index set for all nodes: $N = H \cup C$
L	Index set for load levels
$[a_c, b_c]$	Time window of customer $c \in C$
c^v	Variable costs per time unit
$\hat{c}^{f,h}$	Fixed costs for opening a micro hub
$\hat{c}^{f,b}$	Fixed costs for each cargo bike used
M	Sufficient large number
$[p_l, r_l]$	Mass interval of load level $l \in L$
q_c	Demand mass of node $c \in C$
Q^b	Payload per cargo bike
Q^h_i	Payload per micro hub $i \in H$
s_i	Service time at node $i \in N$
$t_{i,j,l}$	Traveling time dependent on the remaining load level
$f_{i,j}$	Real-value variable indicating the remaining load of a cargo bike traveling from node i to j $(i, j \in N)$
v_i	Real-value variable indicating the arrival time at node $i \in N$
w_i	Integer variable indicating the number of cargo bikes assigned to micro hub $i \in H$
$x_{i,j}$	Binary variable indicating the routing $(i, j \in N)$
y_i	Binary variable indicating if micro hub $i \in H$ is opened
$z_{i,j,l}$	Binary variable indicating if load level l is chosen when traveling from i to j

$f_{i,j} \in \mathbb{R}^+$ displays the remaining bike's load when traveling from i to j, and v_c shows the arrival time at customer c. Table 1 summarizes the used index sets, parameters, and variables.

$$\min \sum_{i\in H} \left(\hat{c}^{f,h} \cdot y_i + \hat{c}^{f,b} \cdot w_i\right) + \sum_{i,j\in N, l\in L} c^v \cdot t_{i,j,l} \cdot z_{i,j,l} \quad (1)$$

subject to

$$\sum_{j\in N} (x_{i,j} - x_{j,i}) = 0 \quad \forall i \in N \quad (2)$$

$$\sum_{j\in N} x_{i,j} = 1 \quad \forall i \in C \quad (3)$$

$$x_{i,j} \le y_i \quad \forall i \in H, j \in C \quad (4)$$

$$\sum_{j\in C} x_{i,j} \le w_i \quad \forall i \in H \quad (5)$$

$$\sum_{j\in N} (f_{j,i} - f_{i,j}) = q_i \quad \forall i \in C \quad (6)$$

$$f_{i,j} \ge q_j \cdot x_{i,j} \quad \forall i \in N, j \in C \quad (7)$$

$$f_{i,j} \le (Q^b - q_i) \cdot x_{i,j} \quad \forall i \in C, j \in N \quad (8)$$

$$\sum_{j\in N} f_{i,j} \le Q^h_i \quad \forall i \in H \quad (9)$$

$$v_i - v_j + s_i + \sum_{l\in L} t_{i,j,l} \cdot z_{i,j,l} \le M \cdot (1 - x_{i,j}) \quad \forall i \in N, j \in C : i \ne j \quad (10)$$

$$a_i \le v_i \le b_i \quad \forall i \in C \quad (11)$$

$$\sum_{l\in L} z_{i,j,l} = x_{i,j} \quad \forall i, j \in N \quad (12)$$

$$\sum_{l\in L} p_l \cdot z_{i,j,l} \le f_{i,j} \le \sum_{l\in L} r_l \cdot z_{i,j,l} \quad \forall i, j \in N \quad (13)$$

$$y_i \in \{0, 1\} \quad \forall i \in H \quad (14)$$

$$w_i \in \mathbb{N} \quad \forall i \in H \quad (15)$$

$$x_{i,j}, z_{i,j,l} \in \{0, 1\} \quad \forall i, j \in N, l \in L \quad (16)$$

$$f_{i,j} \in \mathbb{R}^+ \quad \forall i, j \in N \quad (17)$$

$$v_i \in \mathbb{R}^+ \quad \forall i \in C \quad (18)$$

The objective function minimizes total costs, i.e., fixed costs for hubs and bikes and variable costs for the traveling time. Costs for service are not included, as this is a fixed term and, thus, not relevant for decision-making. Constraints (2) conserve flow, and Constraints (3) ensure that each customer is served exactly once. Micro hubs can only be used if they are opened (Constraints (4)). Constraints (5) determine the number of bikes per hub. The remaining bike's load traveling from i to j is defined in Constraints (6). These constraints additionally eliminate subtours for each bike. The load is at least the customers' demand mass (Constraints (7)) and limited to each bike's payload (Constraints (8)). Constraints (9) limit each hub's capacity. Constraints (10) set the arrival times at customers, taking load-dependent travel times into account. M must be at least the maximum reasonable tour length. Constraints (11) ensure the time windows are adhered to. When traveling from i to j, exactly one load level is selected (Constraints (12)). Constraints (13) ensure that each bike's load is within the load interval. Last, variables are defined.

Instead of minimizing costs for the traveling time, symmetric routing costs based on the distance to travel can be minimized. For this, $\bar{c}^v_{i,j}$ indicates the costs for the traveled distance from i to j $(i, j \in N)$. The load-dependent travel times, then, affect the arrival time at a customer and, thus, if its time window is adhered to.

$$\min \sum_{i\in H} \left(\hat{c}^{f,h} \cdot y_i + \hat{c}^{f,b} \cdot w_i\right) + \sum_{i,j\in N} \bar{c}^v_{i,j} \cdot x_{i,j} \quad (19)$$

5. Adaptive large neighborhood search

This section introduces our ALNS capable of solving larger instances. First, we describe the peculiarity of this ALNS in Section 5.1. Section 5.2 presents an overview of the ALNS. In Section 5.3, we describe how to generate the initial solution, and in Section 5.4 and Section 5.6, we introduce our problem-specific operators. Section 5.5 presents the procedure that decides on the location and fleet. Last, Section 5.7 shows a local search that is included in our ALNS.

5.1. Peculiarity of proposed ALNS algorithm

The ALNS was initially introduced by Røpke and Pisinger (2006) and Pisinger and Røpke (2007) and has been adapted to multiple problem settings, e.g., for the LRP (e.g., Hemmelmayr et al., 2012) or the inventory routing problem (e.g., Aksen et al., 2014).

An ALNS tackling a VRP typically requires only a few operators (four to six) (e.g., Fontaine, 2022; Sacramento et al., 2019). More can be implemented to improve the solution quality (e.g., Pisinger & Røpke, 2007), but are not necessary as a few operators already cover all routing decisions. Considering not only the VRP but also its extensions, e.g., MDVRP, on average, six destroy and 3.7 repair operators are used (Voigt, 2024). Tackling an LRP, the main decisions are not only the operational routing but also the tactical micro hub locations and fleet decisions. Thus, compared to a VRP, there are more decisions to take, resulting in much more complex or more operators being required. This is because there is a need to modify the routing and, additionally, the tactical decisions, which are typically not included in the same operators. So, Hemmelmayr et al. (2012) use twelve, Demir et al. (2012) 17, Koç (2016) ten, and Sirirak and Pitakaso (2018) eleven operators to adjust both the routing and location decision. These huge numbers of operators make it difficult, time-consuming, and extensive to implement. Akpunar and Akpinar (2021) were also aware of this problem separating the operational routing from the tactical decisions. They introduce an ALNS for operational routing (six operators) and apply a variable neighborhood search for the tactical decisions, which, however, requires a further eight operators.

Thus, we develop an ALNS that can solve the LRPTWLTT but can also be applied to the LRP and LRPTW with only four operators that are well and generally functioning and also easy to implement (Voigt, 2024). Note that our operators are, additionally, problem-specific in covering load-dependent travel times and time windows. We can solve the LRPTWLTT by using only four operators as we separate the tactical decision regarding the micro hub locations and the vehicle fleet from the operators, who, therefore, only modify the operational routing. The tactical hub location and fleet decision are then considered by an adaptive procedure executed after applying the destroy and before the repair operators. This adaptive procedure includes innovative features to tackle the new problem introduced. We describe these features in detail in Section 5.5.

5.2. Overview

In each iteration of the ALNS, the operational routing R and potentially the tactical bike and hub set B is adjusted. R includes information on the order of customer visits, and B includes the micro hub locations and the number of cargo bikes per micro hub used. Each bike's load-dependent travel times result from the order of customer visits. Following the idea of, e.g., Vidal et al. (2013), we accept infeasible solutions in each iteration, but infeasibility is penalized in the cost function f dependent on the degree of infeasibility. The solution might be infeasible if a bike's or a hub's capacity is exceeded (c^Q) or the customers' time windows are not adhered to (c^{TW}), i.e., the bike arrives too late at a customer. The cost function is defined as follows, where each "max" function determines the degree of infeasibility.

$$
\begin{aligned}
f(R,B) = {} & \sum_{i\in H}\left(\hat{c}^h\cdot y_i + \hat{c}^b\cdot w_i\right) + \sum_{i,j\in N, l\in L} c^v\cdot t_{i,j,l}\cdot z_{i,j,l} \\
& + c^{TW}\cdot\sum_{i\in C}\max(v_i - b_i, 0) \\
& + c^Q\cdot\sum_{i\in H}\max\Big(\sum_{j\in N} f_{i,j} - Q_i^h, 0\Big) + c^Q\cdot\sum_{i\in H, j\in N}\max(f_{i,j} - Q^b, 0)
\end{aligned}
$$

Algorithm 2: ALNS framework for the LRPTWLTT

1 R, R^{global}, B, B^{global} ← Initial Solution();
  //see Section 5.3
2 **while** *noimprovements* $\leq \alpha$ **do**
3   ChooseOperator();
4   ($\hat{R}$, RemovedCustomers) ← RemoveCustomers(R);
    //see Section 5.4
5   $\hat{B}$ ← RemoveUnusedBikesAndHubs;
6   $\hat{B}$ ← AddingOrShiftingBikeOrHub;
    //see Section 5.5
7   $\hat{R}$ ← RepairCustomers($\hat{R}$,$\hat{B}$, RemovedCustomers);
    //see Section 5.6
8   **if** $f(\hat{R},\hat{B}) < (1+\mu)\cdot f(R^{global}, B^{global})$ **then**
9     $\hat{R}$ ← Local Search($\hat{R}$);
      //see Section 5.7
10   **end**
11   **if** $f(\hat{R},\hat{B}) < f(R^{global}, B^{global})$ **then**
12   (R,B), (R^{global}, B^{global}) ← $(\hat{R},\hat{B})$;
13   **else if** *accept*($f(\hat{R},\hat{B})$, $f(R,B)$) **then**
14   (R,B) ← $(\hat{R},\hat{B})$;
15   UpdateWeights();
16 **end**

Algorithm 2 presents the pseudo-code of our ALNS, which starts with an initial solution (see Section 5.3). Next, in line 2, the while loop runs until the number of consecutive iterations without improvement exceeds a threshold α. In line 3, the destroy and repair operators are chosen (see Sections 5.4 and 5.6). The chosen destroy operator removes at least one customer (line 4). If a bike has no customer left in its route, it is removed. The same applies for hubs if no bike is left at the hub (line 5). Next, in line 6, we apply a problem-specific procedure that decides if a new micro hub is opened, a new bike is added, or a certain number or all bike routes of a single hub are shifted to a different hub. We will describe this procedure in detail in Section 5.5. In line 7, the chosen repair operator inserts the removed customers again. If the solution is promising, i.e., its costs f are below a certain threshold $1+\mu$ of the global best solution (e.g., Hemmelmayr et al., 2012), a local search is executed to improve the routing (lines 8–10). The local search is presented in detail in Section 5.7. The global and current solutions are accepted if the changes lead to a total cost reduction of f, considering penalty costs (lines 11–14). Last, the weights $\eta_{i,j+1}$ for the operators i are updated (line 15) as in Sacramento et al. (2019): $\eta^{op.}_{i,j+1} = \psi\cdot\eta^{op.}_{i,j} + \omega^{op.}\cdot(1-\psi)$ with the reaction factor for the learning curve $\psi\in[0,1]$ and the performance dependent values $\omega\in\mathbb{R}^+$. We apply the same update to the adaptive procedure to add a new bike or hub: $\eta^{proc.}_{j+1} = \psi\cdot\eta^{proc.}_{j} + \omega^{proc.}\cdot(1-\psi)$. Note that if the procedure is not executed, it is not updated, i.e., $\eta^{proc.}_{j+1} = \eta^{proc.}_{j}$.

5.3. Initial solution

One hub, which is geographically best located, is opened. This means that its distance to all customers weighted by the hub's elevation is best. The number of bikes allocated to the hub depends on the total demand mass and the bikes' capacities. Similar to Voigt et al. (2023), the routing is created by applying our "greedy load-dependent insertion" operator (see Section 5.6) to all customers.

5.4. Destroy operators

The destroy operators remove β many customers from routes, with $\beta\in\{1,\dots,\beta^{max}\}$ randomly drawn and β^{max} is set dependent on the number of considered customers (e.g., Sacramento et al., 2019), i.e., $\beta^{max} = \lceil\min(\max(10, 0.2\cdot|C|), 20)\rceil$. Bikes are removed if no customer is left on a route. Further, if no bike is left at a hub, this hub is closed. In detail, we consider the following destroy operators, which are extensions or variations of Hemmelmayr et al. (2012), Fontaine (2022), and Rave et al. (2023a):

Random removal - This operator removes β many customers from any routes, randomly.

Sequence removal - This operator removes β many customers that are served in a sequence. In detail, a random customer is chosen first. Subsequently, $\beta - 1$ many customers who are delivered immediately before or after are removed depending on which customer is closer. The sequence of customers might exceed a single tour. In contrast to a typical "cluster removal" operator, customers are not necessarily geographically adjacent to each other (Voigt, 2024). This operator especially removes bikes and micro hubs from the routing by removing all customers served by a bike, thus making an additional bike removal operator unnecessary - provided that β^{max} exceeds all tour lengths.

5.5. Procedure for adding or shifting a bike or hub

In this section, we describe the procedure that decides if a bike is added, at which hub it is located, and if a bike is shifted to a different hub. Algorithm 3 shows the pseudo-code for this procedure. The decision of whether a bike or a hub is added or shifted is executed based on its historical performance parameter $\eta^{proc.}_j$ in iteration j of the ALNS.

First, if the random number χ^{Add} is below $\eta^{proc.}_j$, a new bike is added to an existing hub, or a new hub is opened and equipped with a single bike (lines 2–4). The new bike's route is empty as customers might be added to this route by our repair operators. The decision of where to place a bike and which hub to open is random but weighted in order to have a good choice. The weight depends on the problem-specific main influencing factors of each hub's load-dependent travel times to the removed customers. Based on preliminary results, we

Algorithm 3: Procedure for adding or shifting a bike or hub

```
1  χ^Add ← random(0,1);
2  if χ^Add ≤ η_j^proc. then
3  |   AddBikeOrHub;
   |   //Add a single bike to existing or new hub
4  end
5  χ^Shift ← random(0,1);
6  if χ^Shift ≤ η_j^proc. then
7  |   χ^ShiftBikeorHub ← random(0,1);
8  |   if χ^ShiftBikeorHub ≤ λ then
9  |   |   for 1...γ^max do
10 |   |   |   ShiftBike;
   |   |   |   //Shift a bike to a different hub
11 |   |   end
12 |   end
13 |   else
14 |   |   ShiftHub;
   |   |   //Shift all bikes of a single hub to a different hub
15 |   end
16 end
```

observed that the hub's elevation and its general accessibility to the removed customers who need to be reinserted are these main factors. The general accessibility of customers is a function of the micro hubs' distance to each removed customer and its demand mass and needs to be determined in each iteration. Additionally, we consider the hub's historical performance, which is updated in each iteration of the ALNS.

Second, a random number of bikes or all bikes of a single hub are shifted to one different hub, which needs to be open (lines 5–16). If a shift is executed (line 6), it is checked next whether a certain number of bikes γ^{max} of any hubs or all bikes of a single hub are shifted to a different hub. γ^{max} many bikes are shifted individually, if $\chi^{ShiftBikeorHub}$ is below a threshold λ (lines 7–12). Else, all bikes of a single hub are shifted (lines 13–15). The maximum number of bike shifts γ^{max} is set as follows: $\gamma^{max} = \lceil \min(\max(5, 0.1 \cdot |C|), 10) \rceil$. The new hub was chosen randomly but weighted by its elevation and general accessibility to all customers. Note that newly opened hubs of the previous steps are also considered.

5.6. Repair operators

The repair operators allocate the removed customers to bikes that are in the solution pool, i.e., the repair operators do not add new hubs or bikes. If a hub or a bike is unused after inserting all customers, it is removed again. For insertion, we consider the following two operators:

Greedy load-dependent insertion - This operator adds the removed customers to routes in the greedy best way, considering load-dependent travel times and time windows.

Greedy load-dependent insertion - with noise - This operator adds a certain noise to the "greedy load-dependent insertion" operator, and each position's cost is adjusted by a factor $\delta \in [0.8, 1.2]$ (Hemmelmayr et al., 2012; Røpke & Pisinger, 2006). This operator increases the number of good and admissible solutions and guarantees an expansion of the solution space through a certain randomness.

5.7. Local search

To improve the routing, we execute a local search for a promising solution. The local search is a two-opt, where each of two customers within any of the bikes' routes is swapped, and it is checked whether this swap leads to an improvement in costs. It follows that the local search does not vary the number of customers served by a bike or from a micro hub; thus, it does not impact the location or fleet decision, i.e., no hubs or bikes are removed or added.

6. Numerical results

In this section, we introduce our numerical setup, test the performance of our ALNS, and present our numerical results. The MILP is implemented in OPL and solved using CPLEX v22.1.1, and the ALNS is implemented in C++. All experiments are conducted on an AMD Ryzen 9 5950X with 128 GB RAM.

6.1. Numerical setup

We consider the VRPTWLTT instances of Fontaine (2022) for five large cities (Fukuoka, Madrid, Pittsburgh, Seattle, and Sydney) with 100 customers and extend them by ten micro hub locations to LRPTWLTT instances. The placement of micro hubs requires not only topographical but also legal features. Since we do not have this information for these cities, we cover the entire area with randomly placed hubs.

The considered cities have relatively large average road gradients of 1.85% (Fukuoka), 2.49% (Madrid), 3.64% (Pittsburgh), 3.91% (Seattle), and 3.60% (Sydney). The customers have a demand mass $q_c \in \{5, 6, \dots, 15\}$ randomly chosen with 10 kg on average, an average time-window length of 52.4 min, and service times of five minutes per customer arise. We consider ten load levels and six different demands as in Fontaine (2022). In Section 6.3.1, we show that ten load levels are sufficient to not lose precision by approximating load-dependent travel times. Each bike has a capacity of $Q^b = 150$ kg, and each hub $i \in H$ of $Q^h_i = 600$ kg. Moreover, each bike has a power of 100 W added to a battery power of 250 W (Fontaine, 2022). We vary the cyclists' power in a sensitivity analysis in Section 6.4.2.

A hub costs 14.07 € per day (Rave et al., 2023a), and a bike 6.48 € per day. Each bike's costs are based on 8000 € for acquisition (Urban Arrow, 2024) with a depreciation time of seven years and expected 10% additional annual insurance, maintenance, and spare part costs. For variable costs, we consider a wage of 15 € per hour.

ALNS parameters are based on pre-testing and aligned with the literature (e.g., Fontaine, 2022; Rave et al., 2023a; Sacramento et al., 2019) and set as follows: The reaction factor for the learning curves of the ALNS ψ is set to 0.85. Starting with an initial weight of $\eta^{op.}_{i,0} = 100$ per operator i, the operators' weights are increased by $\omega^{op.} = 330$ for a global best new solution, adjusted by $\omega^{op.} = 130$ for a local best new solution, and reduced by $\omega^{op.} = 0$ for a worse new solution. Similarly, the initial weight of our procedure for adding or shifting a bike or hub is set to $\eta^{proc.}_0 = 0.5$ and is adjusted by $\omega^{proc.} = 1$, $\omega^{proc.} = 0.5$, or $\omega^{proc.} = 0.2$, dependent on the solution quality. The local search parameter μ is set to 7%. The ALNS stops when there is no improvement in 10,000 runs.

6.2. Benchmark tests

In this section, we test the performance of our ALNS and the chosen operators.

6.2.1. LRPTWLTT benchmark tests

First, we test our ALNS with instances solved by CPLEX within a time limit of one hour. As CPLEX has runtime issues solving the MILP to optimality for instances with 100 customers, we first take a subset of 20 customers and three random micro hub locations. Please note that due to the reduced number of customers, we increase their demand mass to a randomly drawn integer number between 5 and 30, leading to an average demand mass of 18 kg (Fontaine, 2022).

Table 2 presents the aggregated results for each city, i.e., the average of six instances per city. The first column shows the considered city.

Table 2
Performance analysis for small instances (20 customers).

City	Cost objective				CPLEX		Runtime (s)	
	CPLEX	ALNS best	ALNS avg	ALNS σ [%]	Opt	GAP [%]	CPLEX	ALNS
Fukuoka	61.3504	61.3434	61.5043	0.26	1/6	2.75	3340	6
Madrid	61.3123	61.2590	61.3855	0.20	0/6	3.45	3600	7
Pittsburgh	61.4881	61.5114	61.5831	0.11	2/6	2.15	2680	8
Seattle	59.6755	59.6466	59.7430	0.13	0/6	3.88	3600	7
Sydney	59.0134	59.1237	59.1590	0.07	2/6	1.72	2539	7
Average	60.5679	60.5768	60.6750	0.15	5/30	2.79	3152	7

Table 3
Performance analysis for medium instances (50 customers).

City	Cost objective				CPLEX		Runtime (s)	
	CPLEX	ALNS best	ALNS avg	ALNS σ [%]	Opt	GAP [%]	CPLEX	ALNS
Fukuoka	90.8362	53.8371	54.3375	0.73	0/6	45.51	3600	34
Madrid	88.8682	58.7107	59.1821	0.58	0/6	44.91	3600	30
Pittsburgh	88.2808	57.6060	58.1858	0.68	0/6	46.19	3600	32
Seattle	89.8262	51.0603	51.6909	0.88	0/6	51.94	3600	36
Sydney	92.5196	58.6000	59.1098	0.67	0/6	53.31	3600	34
Average	90.0662	55.9628	56.5012	0.71	0/30	48.37	3600	33

The next four columns report the solution found by CPLEX, the best-found solution of five runs of our ALNS, the average found solution, and the standard deviation in percent. These columns report the average values for each of the six instances per city. The next two columns show the number of instances of each city CPLEX has solved to optimality and the average optimality gap in % if no optimality is proven. The last two columns compare the average runtime of CPLEX and running our ALNS.

CPLEX can solve 5 of 30 instances to optimality with an average runtime of 3152 s. For the other instances, the optimality gap is rather low. Our ALNS, on the other hand, finds better solutions on average for the cities of Fukuoka, Madrid, and Seattle despite the low optimality gap to find anything better. Moreover, our ALNS's runtime of a maximum of eight seconds is significantly lower. The results found are stable with $\sigma = 0.15\%$.

Since the solution quality of CPLEX can be interpreted as competitive with these small instances - even if not in the runtime - we consider medium-sized instances in the following. For this, we take the instances as in our case study but reduce the number of customers to 50. All other parameters, e.g., number of load levels and hub locations, stay the same, i.e., $|L| = 10$, $|H| = 10$. Similar to the previous table, Table 3 shows the results.

CPLEX cannot solve a single instance to optimality within 60 min, and the gaps are very large, with 48.4% on average. On the contrary, our ALNS finds better solutions for all instances also being stable with σ of less than 1%. Moreover, our ALNS has an average runtime of only 33 s.

6.2.2. *LRPTW benchmark tests*

To further show our ALNS's solution quality, we benchmark it to the optimally solved LRPTW instances of Ponboon et al. (2016), who extend the well-known Solomon (2005) benchmark instances with additional hub locations. Each instance considers three hub locations. This LRPTW is close to our problem by considering time windows but no load-dependent travel times and a symmetric cost function. Table 4 reports the results with unchanged ALNS parameters. The first column shows the instance names as in Ponboon et al. (2016). The number separated by a hyphen indicates the number of customers. The next columns show the optimal solution, our best solution found, the average solution found out of five runs, the standard deviation in percent, and the runtime in seconds. Last, the table reports the gap Δ between our best-found solution and the optimal solution in percent.

Table 4
Performance analysis for LRPTW instances of Ponboon et al. (2016).

Instance name	Opt. sol.	ALNS				
		Best	Avg.	σ [%]	Runtime [s]	Δ [%]
LRPTW-R101–25	5308	5308	5334.2	0.6	8	0.0
LRPTW-R102–25	5027	5027	5049.8	0.3	5	0.0
LRPTW-R103–25	4294	4294	4309.6	0.7	12	0.0
LRPTW-R104–25	4251	4251	4270.6	0.7	9	0.0
LRPTW-R105–25	4587	4597	4641.8	0.5	9	0.2
LRPTW-R106–25	4438	4438	4453.8	0.3	6	0.0
LRPTW-R107–25	4266	4266	4295.0	0.5	7	0.0
LRPTW-R108–25	4077	4077	4093.4	0.5	7	0.0
LRPTW-R109–25	4299	4299	4387.0	2.4	10	0.0
LRPTW-R110–25	4285	4285	4309.8	0.5	8	0.0
LRPTW-R111–25	4289	4291	4312.2	0.5	10	0.0
LRPTW-R112–25	4250	4253	4302.2	0.7	7	0.1
LRPTW-R101–40	7645	7687	7709.6	0.2	17	0.5
LRPTW-R102–40	7150	7183	7219.2	0.3	24	0.5
LRPTW-R105–40	6919	6981	7043.0	0.6	22	0.9
Average	5005.7	5015.8	5048.8	0.6	11	0.1

While Ponboon et al. (2016) needed nearly eight hours to find the optimal solution for some instances, we found the optimal solution in ten instances with an average runtime of 11 s. For the other instances, the optimality gap is rather small, with always less than 1% and, on average, just 0.1%. In addition, our solutions found are stable with $\sigma < 1\%$ on average. The results are promising as our location procedure and repair operators are developed to consider load-dependent travel times.

6.2.3. *Performance of operators*

In this section, we test the operators we use. Table 5 reports the average number of each operator used to provide a new best solution and the increase in costs if this operator is not included in the ALNS. For this, we consider the LRPTWLTT instances in our case study.

Our "Random removal" and our "Greedy load-dependent insertion" operators are frequently used to find new best solutions. However, our "Sequential removal" operator has a much more significant influence on the final solution found than the "Random removal" has. This could be because the random removal operator only achieves minor improvements - but many of them. In addition, our "Greedy load-dependent insertion - with noise" operator is used much less frequently

Table 5
Average number of operators used to generate the new best solution (1. row), and cost increase if this operator is not included in the ALNS (2. row).

	Destroy operators		Repair operators	
	Random	Sequential	Greedy	Greedy - with noise
#Used operators	105.9	51.2	149.3	7.8
Cost increase	2.1%	11.0%	5.8%	1.6%

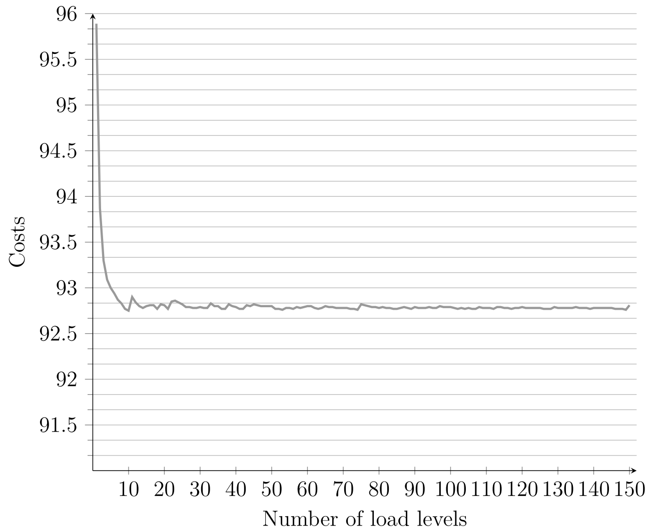

Fig. 1. Objective function values for an increasing number of load levels.

than the one without noise. However, the randomization leads to an improvement of 1.6% in the final solution.

6.3. Numerical tests

In this section, we show that load-dependent travel times can be approximated by load levels without losing precision. Further, we show that our location decision based on a single demand scenario leads to a robust solution - even if the demand varies.

6.3.1. Determining the number of load levels

We now show that in our case study, our chosen number of load levels does not distort our objective function values, as with too few load levels, the nonlinear function of load-dependent travel times is not properly approximated. To determine the loss in precision by the approximation of load levels, we solve a single instance of the city of Seattle with varying load level numbers, starting with one load level and ending with 150 load levels. We chose the city of Seattle as this city has the largest road gradient and, thus, the steepest traveling time curves.

Fig. 1 shows the objective function values for the same solution when one to 150 load levels are considered. While the deviations are rather large, with one to five in comparison to 150, ten load levels only have a deviation in the second decimal place. This indicates that approximating load-dependent travel times by ten load levels leads to a very slight loss of precision. Please note that the solutions are all feasible, i.e., there is no time window break, except if one or two load levels are chosen.

6.3.2. Robustness of location decision

The decision on hub locations is a tactical decision as these are typically operated for a longer time period than a day. This, however, might be problematic as we consider only a single representative demand also used for operational routing planning (Fontaine, 2022). If there are demand fluctuations, this might lead to a significant increase in costs

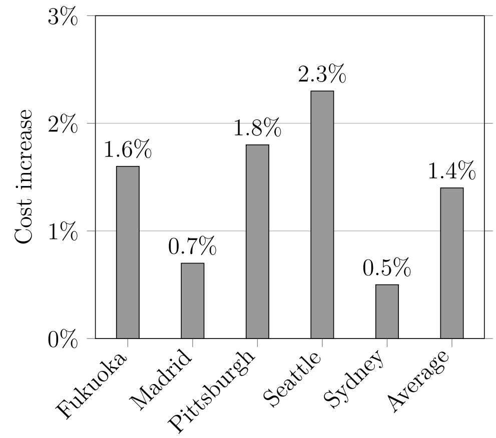

Fig. 2. Cost increase when setting hub locations based on a representative demand.

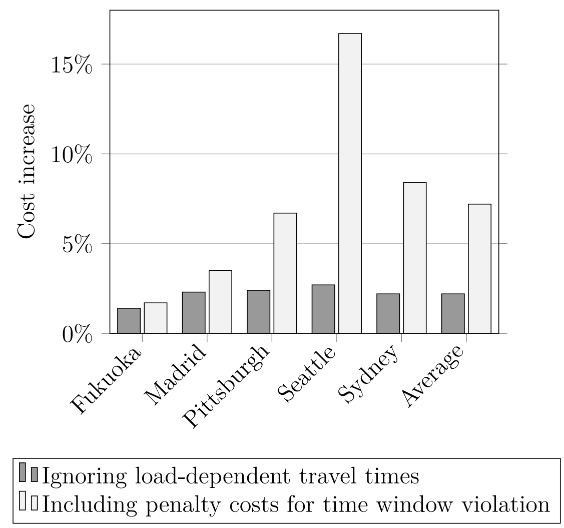

Fig. 3. Cost increase (gray bar) when ignoring load-dependent travel times and when additionally considering penalty costs for time window break (light gray bar).

or even infeasibility due to a time window break. Thus, the question may arise on how robust the found locations are, or in other words, is it required to consider multiple demand scenarios simultaneously? To answer this question, we compute the cost increase when setting the hub locations once for a single demand scenario in comparison to each demand scenario's optimal set hub locations.

For this, we generate a set of instances by varying the number of customers ($|C| \in [50, 100]$) and conduct the following steps. First, we solve an instance with $|C| = 75$ customers. Second, we set the hub locations as in the found solution. Third, we solve an MDVRP with time windows and load-dependent travel times (MDVRPTWLTT) for five instances with $|C| \in [50, 100]$. Fourth, we compare this approach's total costs to the costs when considering the best hub locations for each of the five instances individually (see Fig. 2).

We find that costs increase by 1.4% on average when considering a fixed hub based on a single instance instead of flexible hub locations. This cost increase is relatively low compared to the large variation in the number of customers, which is between −33% and 25%. One main reason for the cost increase is that, in some instances, with fewer customers, a single micro hub instead of two hubs would be optimal. This is, for example, the case in Seattle. It follows that the location decision based on a single representative demand leads to a very minor cost increase despite the strong demand fluctuations.

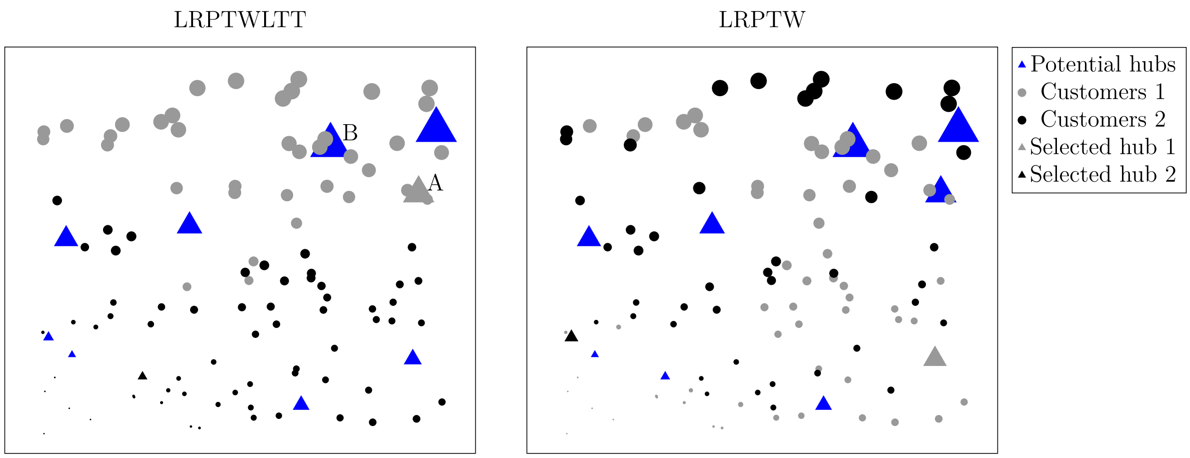

Fig. 4. Customer (points) and hub locations (triangles) when considering (left) and ignoring load-dependent travel times for the city of Seattle. Wider points or triangles indicate a higher elevation.

6.4. Managerial implications

In this section, we show the impact of load-dependent travel times and further perform a sensitivity analysis with increasing cyclists' powers.

6.4.1. Impact of considering load-dependent travel times

We compare the results when solving the LRPTW and the LRPTWLTT to show the impact of considering load-dependent travel times. For this, we evaluate the LRPTW's final solutions with load-dependent travel times. This, however, leads to average time window violations in 18% of all solutions for the cities with a lower road gradient (Fukuoka and Madrid) and in 26% of all solutions for the cities with a larger road gradient (Pittsburgh, Seattle, and Sydney). This infeasibility exists because the load-dependent travel time has the effect of reducing the speed of the bikes, which in turn makes it more difficult to keep to the time windows.

Delayed deliveries may reduce customer satisfaction or increase employee stress. Because of this service decrease and due to the resulting difficulty in comparison, we, thus, add costs of 1€ for each minute the time windows are not adhered to, i.e., each minute the bike arrives too late. Fig. 3 reports the average cost increase in % (gray bar) when ignoring the load-dependent travel times and the average cost increase when there are additional penalty costs for time window breaks (light gray bar). Note that the gray bar only compares feasible solutions, and the light gray bar compares all solutions found.

We find that load-dependent travel times significantly influence the routing and the selected micro hub, resulting in a cost increase of 1.4% in Fukuoka to up to 2.7% in Seattle. This increase is especially larger in cities with a larger elevation on average. Considering penalty costs for time window breaks, costs increase significantly to up to 16.7% with a larger road gradient on average. On the contrary, in Fukuoka, this has nearly no impact on the cost increase.

One reason for the large cost increase - especially in Seattle - is that different hubs are opened, i.e., hubs with a lower elevation if they are closer to customers and that customers are allocated to hubs independent of their elevation. Fig. 4 presents an example hub placement for Seattle. The figure shows the customers' (points) and potential hub locations (triangles) when considering (left) and ignoring load-dependent travel times (right). Broader points or triangles indicate a higher elevation. The opened hubs are dark blue and gray colored. Customers colored gray are served by the gray hub that has a higher elevation, and customers colored in dark blue are served by the dark blue hub with a lower elevation. In this example, the gray hub has an 80.3% higher elevation when considering load-dependent travel times. On the contrary, the second dark blue hub has a similar elevation in both cases. Considering load-dependent travel times, customers with a higher elevation are served by the hub with a higher elevation. Ignoring the load-dependent travel times, customers are assigned to the hubs regardless of their elevation, and thus, more bikes have to climb unfortunate altitudes.

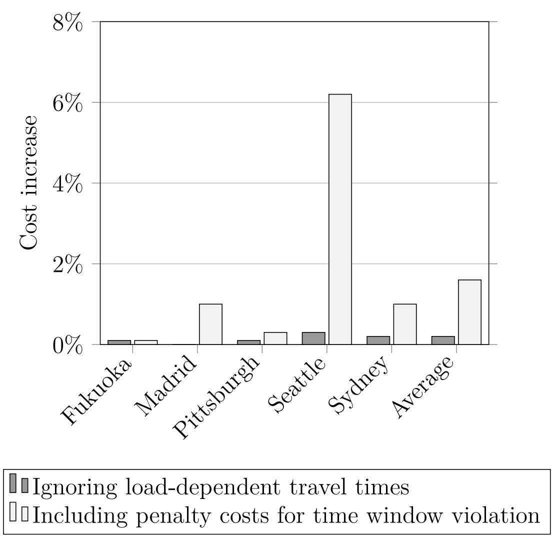

Fig. 5. Cost increase (gray bar) when ignoring load-dependent travel times and when additionally considering penalty costs for time window break (light gray bar) if the cities have a road gradient of 0%.

Considering the LRPTWLTT in Fig. 4, the question may arise as to why hub A is chosen instead of hub B, despite hub B having a higher elevation and being closer to all customers and thus having a preferred position compared to hub A. This is for the following reasons: Time windows and large demands, in particular, force the bikes to give up hub B's advantageous position and instead drive towards hub A because hub A's closest customers have a larger demand and earlier time windows. Therefore, in this case, hub A is advantageous compared to hub B as it is easier to reach these customers, and customers far away additionally have a lower weight. We also tested this and found that, when opening hub B instead of hub A, ceteris paribus costs increase by 1.7%.

Additionally, we analyze the value of load-dependent travel times in flat cities, i.e., with road gradients of 0%. We take the instances for the five considered cities and modify them by setting each customer's and micro hubs' elevation to the same value.

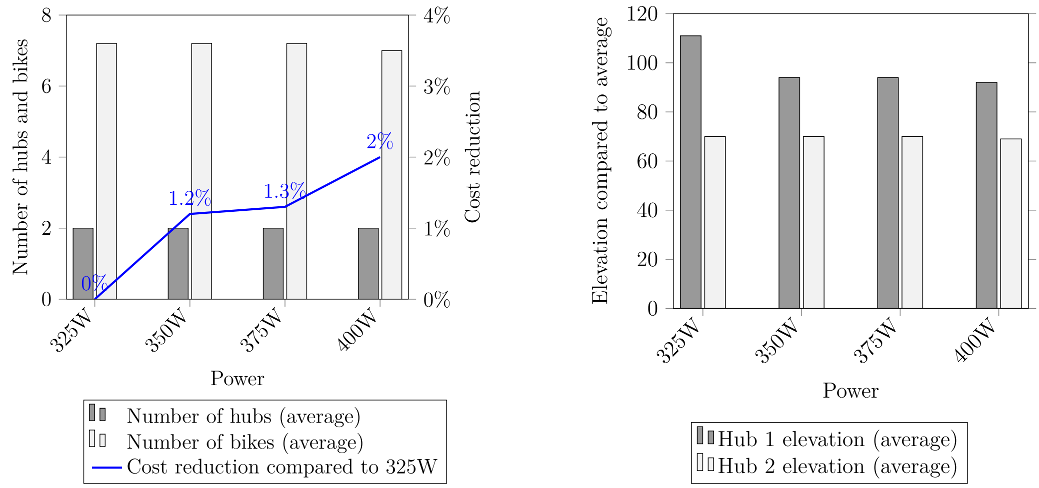

Fig. 6. Development of costs and number of bikes used when increasing the cyclists' power.

Similar to the previous bar chart, Fig. 5 shows the cost increase (gray bar) when ignoring load-dependent travel times and when additionally considering penalty costs for time window break (light gray bar) if the cities have a road gradient of 0%.

We find that cost savings by considering load-dependent travel times decrease when there is no street gradient. However, in 13% of all instances, ignoring load-dependent travel times still leads to infeasible solutions. Considering load-dependent travel times leads to average improved results of 0.2% or 1.6% when including penalty costs, respectively since the degree of infeasibility is less compared to non-flat cities. The time window violations are very small in the cities of Fukuoka and Pittsburgh and a bit larger in Madrid and Sydney. However, there is a large time window violation in Seattle. This outlier shows that there can be considerable differences even in flat cities, as hubs are then set unfortunately: two hubs were chosen that were very off-center and close to each other.

6.4.2. Impact of cyclists' power

In this section, we vary the cyclists' power. Powers are set as in Fontaine (2022): P = 325 W, 350 W, 375 W, and 400 W. These powers include the battery's power of 250 W. Fig. 6 shows the development of costs and bike and hub numbers with increasing power (left) as well as the development of the average elevation of the selected hubs (right). Note that in each instance, two hubs are opened.

With increasing power, the costs and the number of bikes decrease as the traveling times decrease, and more customers can be served during a tour. As in Fig. 4, the hubs can be clustered in two types of elevation levels, a first (hub 1) with a larger elevation and a second (hub 2) with a lower elevation. Considering lower cyclists' power, the hub 1 location has a slightly higher elevation than average. This allows customers in the valley to be supplied from the potentially closer hub 2, while customers at higher altitudes are more likely to be supplied from hub 1. However, increasing cyclists' power makes a lower elevation more advantageous, as the higher positions are usually at the edge of the city and not in its center. Hub 2's elevation, on the other hand, remains unchanged.

7. Conclusion

This paper introduces the location routing problem with load-dependent travel times and time windows, where a larger bike's load increases its travel time. We formulate the problem as MILP and present an ALNS with a problem-specific procedure for the location and fleet decision and problem-specific operators. This allows for the use of only a few operators that just modify the bikes' routing without any influence on the tactical location and fleet decisions, making it less time-consuming to implement. We show the performance of the ALNS with optimally solved instances for our problem setting and LRPTW benchmark instances. We find that the approximation of load-dependent travel times via load levels only leads to a minimal loss in precision. Further, we show the robustness of our found location decision based on a representative demand and generate multiple managerial insights based on instances for the cities of Fukuoka, Madrid, Pittsburgh, Seattle, and Sydney.

In our numerical study, we find that ignoring load-dependent travel times, costs increase by 2.2% on average. Additionally, routing becomes infeasible in 18%–26% of the considered instances (dependent on the street gradient) as time windows are not adhered to. If cities' streets have no gradient, ignoring load-dependent travel times still leads to exceeding time windows in 13% of all instances.

Future research may apply our methodology to a dataset with real micro hub locations, which we do not consider due to regulatory conditions for the micro hub placements. Thus, this might be a limitation to our results. Further, applications of cargo bikes in Quick Commerce Xufei et al. (2024) may be included, or integration into a two-echelon structure, i.e., the hubs' supply from a central depot, is also optimized. This is, however, only relevant if the area to cover is larger, e.g., in suburban areas. Further, a consideration of mobile instead of fixed hubs can be considered. This, however, requires synchronization between both echelons as only a single bike's load is transported to a hub. Finally, the problem could also be extended to include soft time windows or even costs for failed delivery.

CRediT authorship contribution statement

Alexander Rave: Writing – original draft, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Pirmin Fontaine:** Writing – review & editing, Visualization, Supervision, Resources, Project administration, Methodology, Investigation, Conceptualization.

References

Akpunar, Ö. Ş., & Akpinar, Ş. (2021). A hybrid adaptive large neighbourhood search algorithm for the capacitated location routing problem. *Expert Systems with Applications*, *168*, Article 114304.

Aksen, D., Kaya, O., Sibel Salman, F., & Tüncel, Ö. (2014). An adaptive large neighborhood search algorithm for a selective and periodic inventory routing problem. *European Journal of Operational Research*, *239*(2), 413–426.

Amazon (2024). Amazon opens micromobility hub in berlin for e-cargo bike deliveries. World Wide Web, URL https://www.aboutamazon.eu/news/sustainability/amazon-opens-micromobility-hub-in-berlin-for-e-cargo-bike-deliveries. (Last Access: 26 September 2024).

Barreto, S., Ferreira, C., Paixão, J., & Santos, B. S. (2007). Using clustering analysis in a capacitated location-routing problem. *European Journal of Operational Research*, *179*(3), 968–977.

Bektaş, T., & Laporte, G. (2011). The pollution-routing problem. *Transportation Research, Part B (Methodological)*, *45*(8), 1232–1250.

Berger, R. T., Coullard, C. R., & Daskin, M. S. (2007). Location-routing problems with distance constraints. *Transportation Science*, *41*(1), 29–43.

Bombach, E. (2016). *Traktionskonzept für ein Lastenfahrrad: Untersuchung von Fahrverhalten und Antriebsstrang*. Springer Fachmedien Wiesbaden.

Boysen, N., Fedtke, S., & Schwerdfeger, S. (2021). Last-mile delivery concepts: a survey from an operational research perspective. *OR Spectrum*, *43*(1), 1–58.

Brandstätter, C., & Reimann, M. (2018). The line-haul feeder vehicle routing problem: Mathematical model formulation and heuristic approaches. *European Journal of Operational Research*, *270*(1), 157–170.

Bruni, M. E., Khodaparasti, S., & Perboli, G. (2023a). The drone latency location routing problem under uncertainty. *Transportation Research Part C (Emerging Technologies)*, *156*, Article 104322.

Bruni, M. E., Khodaparasti, S., & Perboli, G. (2023b). Energy efficient UAV-based last-mile delivery: A tactical-operational model with shared depots and non-linear energy consumption. *IEEE Access*, *11*, 18560–18570.

Canca, D., De-Los-Santos, A., Laporte, G., & Mesa, J. A. (2019). Integrated railway rapid transit network design and line planning problem with maximum profit. *Transportation Research Part E: Logistics and Transportation Review*, *127*, 1–30.

Cheng, C., Adulyasak, Y., & Rousseau, L.-M. (2020). Drone routing with energy function: Formulation and exact algorithm. *Transportation Research, Part B (Methodological)*, *139*, 364–387.

Demir, E., Bektaş, T., & Laporte, G. (2012). An adaptive large neighborhood search heuristic for the pollution-routing problem. *European Journal of Operational Research*, *223*(2), 346–359.

DHL (2020). DHL and reef technology launch pilot to use ecofriendly cargo bikes for deliveries in downtown Miami. World Wide Web, URL https://www.dhl.com/us-en/home/press/press-archive/2020/dhl-and-reef-technology-launch-pilot-to-use-ecofriendly-cargo-bikes-for-deliveries-in-downtown-miami.html. (Last Access: 26 September 2024).

DHL (2021). DHL supply chain trials ecargo bike for home deliveries. World Wide Web, URL https://www.dhl.com/gb-en/home/press/press-archive/2021/dhl-supply-chain-trials-ecargo-bike-for-home-deliveries.html. (Last Access: 26 September 2024).

Drexl, M., & Schneider, M. (2015). A survey of variants and extensions of the location-routing problem. *European Journal of Operational Research*, *241*(2), 283–308.

Dukkanci, O., Kara, B. Y., & Bektaş, T. (2021). Minimizing energy and cost in range-limited drone deliveries with speed optimization. *Transportation Research Part C (Emerging Technologies)*, *125*, Article 102985.

Fontaine, P. (2022). The vehicle routing problem with load-dependent travel times for cargo bicycles. *European Journal of Operational Research*, *300*(3), 1005–1016.

Hemmelmayr, V. C., Cordeau, J.-F., & Crainic, T. G. (2012). An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. *Computers & Operations Research*, *39*(12), 3215–3228.

Huang, Y.-H., Blazquez, C. A., Huang, S.-H., Paredes-Belmar, G., & Latorre-Nuñez, G. (2019). Solving the feeder vehicle routing problem using ant colony optimization. *Computers & Industrial Engineering*, *127*, 520–535.

Jeong, H. Y., Song, B. D., & Lee, S. (2019). Truck-drone hybrid delivery routing: Payload-energy dependency and no-fly zones. *International Journal of Production Economics*, *214*, 220–233.

Kancharla, S. R., & Ramadurai, G. (2020). Electric vehicle routing problem with non-linear charging and load-dependent discharging. *Expert Systems with Applications*, *160*, Article 113714.

Koç, Ç. (2016). A unified-adaptive large neighborhood search metaheuristic for periodic location-routing problems. *Transportation Research Part C (Emerging Technologies)*, *68*, 265–284.

Kramer, R., Subramanian, A., Vidal, T., & dos Anjos F. Cabral, L. (2015). A matheuristic approach for the pollution-routing problem. *European Journal of Operational Research*, *243*(2), 523–539.

Lin, S.-W., Yu, V. F., & Chou, S.-Y. (2009). Solving the truck and trailer routing problem based on a simulated annealing heuristic. *Computers & Operations Research*, *36*(5), 1683–1692, Selected papers presented at the Tenth International Symposium on Locational Decisions (ISOLDE X).

Logistra (2022). UPS feiert zehn Jahre Lastenradeinsatz. World Wide Web, URL https://logistra.de/news/nfz-fuhrpark-lagerlogistik-intralogistik-ups-feiert-zehn-jahre-lastenradeinsatz-181469.html. (Last Access: 26 September 2024).

Maghfiroh, M. F. N., Yu, V. F., Redi, A. A. N. P., & Abdallah, B. N. (2023). A location routing problem with time windows consideration: A metaheuristics approach. *Applied Sciences*, *13*(2).

Mühlbauer, F., & Fontaine, P. (2021). A parallelised large neighbourhood search heuristic for the asymmetric two-echelon vehicle routing problem with swap containers for cargo-bicycles. *European Journal of Operational Research*, *289*(2), 742–757.

Murray, C. C., & Chu, A. G. (2015). The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C (Emerging Technologies)*, *54*, 86–109.

Nishira, M., Ito, S., Nishikawa, H., Kong, X., & Tomiyama, H. (2023). An integer programming based approach to delivery drone routing under load-dependent flight speed. *Drones*, *7*(5).

Pisinger, D., & Røpke, S. (2007). A general heuristic for vehicle routing problems. *Computers & Operations Research*, *34*(8), 2403–2435.

Ponboon, S., Qureshi, A. G., & Taniguchi, E. (2016). Branch-and-price algorithm for the location-routing problem with time windows. *Transportation Research Part E: Logistics and Transportation Review*, *86*, 1–19.

Rave, A., Fontaine, P., & Kuhn, H. (2023a). Drone location and vehicle fleet planning with trucks and aerial drones. *European Journal of Operational Research*, *308*(1), 113–130.

Rave, A., Fontaine, P., & Kuhn, H. (2023b). Drone network design for emergency resupply of pharmacies and ambulances. Available at SSRN 4569199.

Røpke, S., & Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, *40*(4), 455–472.

Rothenbächer, A.-K., Drexl, M., & Irnich, S. (2018). Branch-and-price-and-cut for the truck-and-trailer routing problem with time windows. *Transportation Science*, *52*(5), 1174–1190.

Sacramento, D., Pisinger, D., & Røpke, S. (2019). An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. *Transportation Research. Part C: Emerging Technologies*, *102*, 289–315.

Salhi, S., & Rand, G. K. (1989). The effect of ignoring routes when locating depots. *European Journal of Operational Research*, *39*(2), 150–156.

Sarbijan, M. S., & Behnamian, J. (2022). Real-time collaborative feeder vehicle routing problem with flexible time windows. *Swarm and Evolutionary Computation*, *75*, Article 101201.

Schiffer, M., & Walther, G. (2017). The electric location routing problem with time windows and partial recharging. *European Journal of Operational Research*, *260*(3), 995–1013.

Schiffer, M., & Walther, G. (2018). An adaptive large neighborhood search for the location-routing problem with intra-route facilities. *Transportation Science*, *52*(2), 331–352.

Sirirak, W., & Pitakaso, R. (2018). Marketplace location decision making and tourism route planning. *Administrative Sciences*, *8*(4), URL https://www.mdpi.com/2076-3387/8/4/72.

Solomon, M. M. (2005). VRPTW benchmark problems. *World Wide Web*, URL http://web.cba.neu.edu/~msolomon/problems.htm. (Last Access: 05 March 2024).

Tamke, F., & Buscher, U. (2023). The vehicle routing problem with drones and drone speed selection. *Computers & Operations Research*, *152*, Article 106112.

Tuzun, D., & Burke, L. I. (1999). A two-phase tabu search approach to the location routing problem. *European Journal of Operational Research*, *116*(1), 87–99.

uit het Broek, M. A. J., Schrotenboer, A. H., Jargalsaikhan, B., Roodbergen, K. J., & Coelho, L. C. (2021). Asymmetric multidepot vehicle routing problems: Valid inequalities and a branch-and-cut algorithm. *Operations Research*, *69*(2), 380–409.

Urban Arrow (2024). Cargo - urban arrow. *World Wide Web*, URL https://urbanarrow.com/product/bikes/business/cargo/. (Last Access: 14 Februrary 2024).

Vidal, T., Crainic, T. G., Gendreau, M., & Prins, C. (2013). A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & Operations Research*, *40*(1), 475–489.

Voigt, S. (2024). A review and ranking of operators in adaptive large neighborhood search for vehicle routing problems. *European Journal of Operational Research*.

Voigt, S., Frank, M., Fontaine, P., & Kuhn, H. (2022). Hybrid adaptive large neighborhood search for vehicle routing problems with depot location decisions. *Computers & Operations Research*, *146*, Article 105856.

Voigt, S., Frank, M., Fontaine, P., & Kuhn, H. (2023). The vehicle routing problem with availability profiles. *Transportation Science*, *57*(2), 531–551.

Wilson, D. G., & Schmidt, T. (2020). *Bicycling science*. MIT Press.

Wu, Z., Wang, J., Chen, C., & Liu, Y. (2023). The load-dependent electric vehicle routing problem with time windows. *International Journal of Shipping and Transport Logistics*, *17*(1–2), 182–213.

Xia, Y., Zeng, W., Zhang, C., & Yang, H. (2023). A branch-and-price-and-cut algorithm for the vehicle routing problem with load-dependent drones. *Transportation Research, Part B (Methodological)*, *171*, 80–110.

Xufei, Y., Ostermeier, M., & Hübner, A. (2024). Winning the race to customers with micro-fulfillment centers: an approach for network planning in quick commerce. *Central European Journal of Operations Research*.