



Delft University of Technology

Multi-purpose pickup and delivery problem for combined passenger and freight transport

Hatzenbühler, Jonas; Jenelius, Erik; Gidófalvi, Győző; Cats, Oded

DOI

[10.1007/s11116-024-10482-9](https://doi.org/10.1007/s11116-024-10482-9)

Publication date

2024

Document Version

Final published version

Published in

Transportation

Citation (APA)

Hatzenbühler, J., Jenelius, E., Gidófalvi, G., & Cats, O. (2024). Multi-purpose pickup and delivery problem for combined passenger and freight transport. *Transportation*, 52(5), 1975-2006.
<https://doi.org/10.1007/s11116-024-10482-9>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.



Multi-purpose pickup and delivery problem for combined passenger and freight transport

Jonas Hatzenbühler¹ · Erik Jenelius¹ · Győző Gidófalvi² · Oded Cats^{1,3}

Accepted: 21 March 2024
© The Author(s) 2024

Abstract

Recent advances in the development of modular transport vehicles allow deploying multi-purpose vehicles, which enable alternate transport of different demand types. In this study, we propose a novel variant of the pickup and delivery problem, the multi-purpose pickup and delivery problem, where multi-purpose vehicles are assigned to serve a multi-commodity flow. We solve a series of use case scenarios using an exact optimization algorithm and an adaptive large neighborhood search algorithm. We compare the performance of a multi-purpose vehicle fleet to a mixed fleet of single-purpose vehicles. Depending on cost parameters, our findings suggest that in certain scenarios, the total costs can be reduced by an average of 13% when multi-purpose vehicles are deployed, while at the same time reducing total vehicle trip duration and total distance traveled by on average 33% and 16%, respectively. The required fleet size can be reduced by 35% on average when operating multi-purpose vehicles. The results can be used by practitioners and policymakers to determine if the combined service of passenger and freight demand flows with multi-purpose vehicles in a given system will yield benefits compared to existing transport operations.

Keywords Public transportation · Freight transportation · Multi-purpose vehicles · Heuristic optimization

✉ Jonas Hatzenbühler
jonas.hatzenbuehler@gmx.de

Erik Jenelius
erik.jenelius@abe.kth.se

Győző Gidófalvi
gyozo@kth.se

Oded Cats
O.Cats@tudelft.nl

¹ Division of Transport Planning, KTH Royal Institute of Technology, Stockholm, Sweden

² Division of Geoinformatics Integrated Transport Research Lab (ITRL), KTH Royal Institute of Technology, Stockholm, Sweden

³ Department of Transport and Planning, Delft University of Technology, Delft, Netherlands

Highlights

- proposing and solving the novel multi-purpose pickup and delivery problem
- analysis of urban scenarios with varied spatial and temporal demand
- cost savings by an average of 13%
- reduction of fleet size by an average of 35%
- reduction of total vehicle trip duration by an average of 33%
- reduction of total distance travelled by an average of 16%

Introduction

The ongoing trends of urbanization, increasing e-commerce and digitalization of supply chain management challenge the efficiency and sustainability of existing transportation systems. Savelsbergh and Woensel (2016) and Los et al. (2020) stress the importance of integrated and collaborative transportation solutions to transform current systems to be more efficient and sustainable. The authors specifically address the important role played by multi-purpose vehicles in such a transformation.

Public transportation and urban freight delivery operations are conventionally designed as two independent systems. Public transportation systems are primarily designed to satisfy passenger demand (Ceder and Wilson 1986), while logistic operations are primarily designed to meet delivery times (Ghilas et al. 2016b). The separate planning of both systems [passenger and freight] may lead to inefficiencies in both systems. One approach to improve the planning of transportation systems is to consider the consolidation of multiple demand flows, simultaneously serving both systems using multi-purpose vehicles. In this work we focus on demand-responsive transportation systems and analyse their potential when consolidating passenger and freight demand.

The concept of combining multiple demand flows in one transportation system is known as integrated transportation. In the past, several projects have investigated and demonstrated the successful integration of passenger and freight transportation in urban environments. In Amsterdam, Netherlands (Marinov et al. 2013) a pilot project investigated the delivery of consumer goods in specially designed trams. These trams transported goods from a suburban depot to a shopping mall in the city center. The integration of flows was achieved by sharing the same track network and infrastructure as the passenger trams. The service was operating from 2007-2009 but had to be terminated since the operations could not be realized without political and financial support. Another example of integrated transportation is the supply of a vehicle production plant in Dresden, Germany (Mueller-Eberstein and Franke 2000). For 19 years, a tram operated there during off-peak hours between the production plant and at the logistic center and used the same tracks as the passenger tram.

The consolidation of multiple demand flows can be realized in different ways (Mourad et al. 2019). One concept is sequential integration, where different demand types are transported by the same vehicle but during different times. Another concept is the simultaneous integration, where different demand types can be transported by the same vehicle at the same time. Compared to simultaneous integration concepts, sequential integration requires vehicles to be physically adjusted for each item type. However, the combined optimization of multiple item types may still lead to an overall improved system compared to the independent operation of single-purpose fleets. Additionally, different types of demand are not mixed, i.e., passenger requests are not mixed with freight items. This guarantees that the level of service

for passengers is not compromised by the additional freight demand. Unlike for simultaneous integration concepts, freight requests are not mixed with passenger requests and do not lead to additional stops for the vehicles.

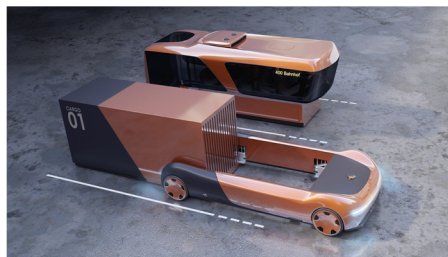
In this study, we extend the pickup and delivery problem formulation to facilitate the operation of multi-purpose vehicles, i.e., we propose the Multi-Purpose Pickup and Delivery Problem (MP-PDP). In a series of experiments we analyze the potential benefits of this vehicle concept over conventional systems. The proposed problem is a novel variation of the heterogeneous routing problem (see Koç et al. 2016) and the truck-and-trailer routing problem (see Derigs et al. 2013). The vehicle operations can be explained as follows: A vehicle operates on a route that starts and ends at a depot. A vehicle can either deliver goods from the depot to a set of customers or transport passengers from their origins to their destinations. A vehicle consists of two parts. The first part is the platform, which contains steering, engine, and wheels. The second part is a functional module that determines the purpose of the vehicle - either passenger transportation or freight delivery module. The module can be exchanged at a depot or a special service depot, and thus change the purpose of the vehicle (see Fig. 1). In the proposed problem two different modules can be used, one exclusively for passenger transport (bus/ride-pooling operations) and the other for the exclusive delivery of freight items (freight truck operation).

The contribution of this paper is threefold. First, the work addresses the research gap of sequentially integrated transportation systems for combined passenger and freight transportation on road networks. Hence, our work extends the sequential integration concepts for road network-based operations by adding the concept of modularity and vehicle module handling to the problem formulation. Second, the proposed problem formulation extends the existing research on PDP by separating the assignment of platforms and modules to requests, which allows the modelling of multiple changing processes at different locations throughout a single route. For these problem-specific model characteristics additional heuristics are implemented. This extends the truck-and-trailer routing problem in which the trailer is dropped and connected to the truck at the same location. Third, the paper studies the impact of multi-purpose vehicles through analyzing various large-scale scenarios.

The potential implications of the conducted research are mainly twofold. First, through the consolidation of passenger and freight demand the level of service for both can be improved. This implies that a transportation system which is designed considering multiple demand types simultaneously can be superior to individually designed transportation systems. Second, reusing vehicle platforms can increase the utilization of high cost vehicle supply, which can allow for lower operation and investment costs compared to individually designed transportation systems.



(a) Scania NXT vehicle concept (Scania, 2020)



(b) U-Shift vehicle concept (DLR, 2021)

Fig. 1 Illustration of the multi-purpose vehicle concept

The remainder of the paper is structured as follows. First, the relevant literature is reviewed in “[Literature review](#)”. Then the problem formulation and methodology of this study are described in “[Methodology](#)”. The experimental design and the results are discussed in “[Experimental design](#)” and “[Results](#)”, respectively. The paper closes with a critical assessment of the results, a conclusion, and an outlook for potential future studies in “[Discussion and conclusion](#)”.

Literature review

The efficient planning of multiple vehicle routes is first envisioned and described in Dantzig and Ramser (1959) as the truck dispatching problem. Since then numerous variations, extensions, and solution algorithms have been developed (see Toth and Vigo 2014; Koç et al. 2016; Dünder et al. 2021 for recent comprehensive literature reviews). Two of the most studied variations of the original problem formulation are the Pickup and Delivery Problem, where each request has a pickup and delivery location; and the more general Dial-a-ride Problem (DARP), in which routes are planned for on-demand requests (see Desrochers et al. 1988; Savelsbergh and Sol 1995, and Cordeau and Laporte 2007; Berbeglia et al. 2010). For this work the following four VRP variations are most relevant. First, the heterogeneous vehicle routing problem which is comprehensively reviewed in Koç et al. (2016); second, the swap body vehicle routing problem (VeRoLog 2014); third, the trailers and transshipment vehicle routing problem (Drexel 2013); and fourth, the share-a-ride problem. We elaborate on the relation between the problem addressed in this study and each of the related problems in the subsequent paragraphs.

The heterogeneous routing problem describes logistic/routing operations with different types of demand and demand type specific vehicles. A common practical application of this routing problem is found in healthcare transportation. Parragh (2011) and Parragh et al. (2012) describe a DARP with heterogeneous demand and vehicles for the transportation of patients. In their model vehicles may be equipped with staff seats, patient seats, stretchers and wheelchairs which in turn define the demand type and capacity for each vehicle. Another form of heterogeneous routing problems is considered by Rekiek et al. (2006) and Melachrinoudis et al. (2007), where vehicles with different capacities are utilized to serve a single type of demand. This problem is closely related to mixed vehicle routing problems, which do not consider pickup and drop off positions for each request. In comparison to these heterogeneous routing problems the model proposed in this work allows for an en-route change of the vehicle configuration. In the previously studied heterogeneous routing problems, the vehicle configuration is decided upon before the depot departure and remains unchanged until the vehicle returns to the depot. Additionally, the number of configuration changes is not limited, allowing for several different configurations for a given route. In Qu and Bard (2013) and Tellez et al. (2018) the authors present heterogeneous PDP and DARP with configurable vehicles, respectively. In these works vehicles can reconfigure their interior and, by that, change the capacity of the vehicle. The authors propose a mixed-integer program which is solved using an adaptive large neighborhood search (ALNS) algorithm. In Qu and Bard (2013) the authors analyze several scenarios and conclude that cost savings of 30%-40% can be achieved by changing the configuration of the vehicles.

The Swap Body Vehicle Routing Problem (SB-VRP) was introduced in a computation challenge and several research teams have developed solution algorithms, for example Huber and Geiger (2017), Todosijević et al. (2017) and Toffolo et al. (2018). The problem considers

the routing of trucks, which can attach or remove trailers of a certain length. The nature of this problem is similar to MP-PDP proposed here. However, in the SB-VRP the start and end location of a trailer has to be one and the same depot, whereas in the MP-PDP a module can be loaded and dropped at any depot or service depot. Furthermore, no multi-depot functionality is implemented in the SB-VRP problem formulation. Additionally, the SB-VRP can deal with only two different types/sizes of trailers whereas the proposed problem formulation can consider more vehicle types, e.g., passenger, freight, and waste transportation. Finally, the proposed vehicle routing problem extends the SB-VRP by adding additional constraints, such as maximum range per platform.

The third group of related vehicle routing problems are trailer and transshipment problems (Drexel 2013). In addition to an adjusted objective function formulation, several additional constraints capturing the multi-depot considerations in the proposed MP-PDP formulation create a new problem variant. In truck-and-trailer routing problems only freight demand is considered (Derigs et al. 2013 and Parragh and Cordeau 2017). Moreover, the types of trailer are limited to one; hence, the model is limited to consider hooking and unhooking of trailers. In contrast, the MP-PDP allows for the investigation of several demand type-specific modules, with different capacities.

Li et al. (2014) investigate if another mode of passenger transportation, namely private taxi rides, can be used for integrated urban transportation. The authors propose a reduced version of the Share-a-Ride problem and the Freight Insertion problem. The problem minimizes the additional operating costs of adding freight items in a set of planned taxi trips. The authors solve their proposed mixed-integer linear program for static and dynamic demand scenarios. The numerical results are sensitive to the spatial distribution of the freight demand. The authors conclude that the integration of freight items into taxi services is a promising solution for urban areas. However, this new integrated mode should be complemented by a traditional truck service to guarantee the delivery of all packages. In their problem the authors consider the taxi trips as pre-planned and identify how to best insert freight requests into these. By this the authors limit the problem complexity to an insertion problem which does not consider the simultaneous planning of already existing routes. In our proposed problem routes and different demand types are planned for simultaneously. Schröder and Liedtke (2017) present a multi-agent simulation model for passenger and freight transportation. They investigate the impacts of various policy measures, i.e. special vehicle tolls.

In a related study by Hatzenbühler et al. (2023), the authors propose an extension of the PDP using modular multi-purpose vehicles which can serve multiple demand types simultaneously by forming platoons. The authors analyze several scenarios in Stockholm, Sweden and could find cost savings of over 50% when compared to traditional vehicle operations. In addition empty vehicle kilometers could be reduced by over 60% in some scenarios.

In contrast to this work, in their model all vehicles can be driven individually and transport either passenger or freight requests. In contrast to this paper a vehicle can never change what type of request it can serve. In this paper the same vehicle can i.e. transport passengers at the beginning of its route and freight at the end of its route. This is a fundamental difference which has consequences for modelling and applications. In addition, multiple vehicles can connect to each other to form platoons. These platoons remain unchanged for the entire trip but can be freely re-configured (number of vehicles and types of vehicles) at each depot. In this paper vehicles cannot form platoons and always operate as individual vehicles. Finally, the vehicles do not consist of a platform and a module but rather just the vehicle, hence the range and cost of the vehicles are modelled differently.

Since the computational complexity of vehicle routing and scheduling problems has proven to be NP-hard (Lenstra and Kan 1981), it is challenging to efficiently solve these

problems for larger scenarios. Two general approaches are typically used to solve the VRP and its variants: (i) the utilization of exact algorithms (e.g. Branch-and-Cut, Branch-and-Bound) and, (ii) the development of problem-specific heuristics or meta-heuristic algorithms (e.g. Simulated Annealing, Artificial Bee Colony, Genetic Algorithm or Large Neighborhood Search). In Arslan et al. (2016) a crowd-sourced delivery system for parcels and passengers is solved using an exact solution approach. An exact algorithm for the shared ride problem is presented by Beirigo et al. (2018), while Ghilas et al. (2016b) solve the PDP with time windows and scheduled lines using CPLEX.

Due to the computational complexity of VRP problems, most researchers implement heuristic algorithms to solve large problems. Chew et al. (2013) develop a bi-objective genetic algorithm (GA) for the VRP and show its ability to reach improved objective values over previously published algorithms for Mandl's benchmark problems. Alizadeh Foroutan et al. (2020) apply a similar genetic algorithm as well as simulated annealing (SA) algorithm to the green routing problem. The authors show that GA converges faster, whereas SA results in better solution robustness and qualities. In Ropke and Pisinger (2006) the ALNS is presented. It is shown that the ALNS improves the best known solutions for VRP problems by around 50%. Additional computational experiments indicate convergence robustness and its adaptability to various problem formulations. These results are confirmed in a later study by Pisinger and Røpke (2010) which shows that large-scale neighborhood search methods lead to fast and robust convergence for complex combinatorial problems. The authors propose variable local search algorithms and adaptive neighborhood definitions to further improve the computational efficiency of the algorithm. In the works of Masson et al. (2013), Ghilas et al. (2016a) and Li et al. (2016) the authors apply the ALNS to a variety of VRPs.

Based on the reviewed literature, we conclude that in recent years the integration and combination of multiple demand types in different transportation modes have been investigated. Studies have shown benefits of simultaneous combination problems. Several authors proposed novel meta-/heuristic algorithms to solve complex VRP problems and showed their superiority in computation time and objective value over exact algorithms. Therefore, we have used a meta-heuristic optimization algorithm to solve the novel MP-PDP.

Methodology

The main characteristics of the pickup and delivery problem as proposed in this work is the multi-purpose vehicle concept and the consolidation of passenger and freight requests. In this section, the detailed problem formulation and solution algorithm are presented.

Proof of concept

In Fig. 2 a simple example is given to showcase the theoretical benefits of multi-purpose vehicles over conventional vehicles. Figure 2a shows the conventional case. The solution is optimal and utilizes two vehicles, one serving the freight requests (blue), while the second vehicle serves the passenger requests (red). Therefore, this solution uses two platforms and two modules to serve the requests. In Fig. 2b the same demand is served on a single route (blue). This solution utilizes the additional service depot to change the module and hence the purpose of the vehicle. Therefore, this solution uses one platform and two modules. The number of platforms is thus reduced at the cost of exchanging modules once. Additionally, the total vehicle operation time is reduced from $20min + 20min + 30min + 30min + 40min +$

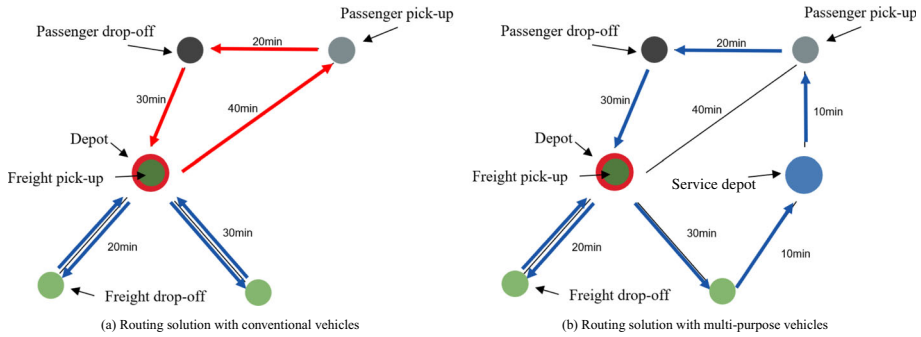


Fig. 2 An illustration of conventional vehicle operations and multi-purpose vehicle operations

$20min + 30min = 190min$ to $20min + 20min + 30min + 10min + 10min + 20min + 30min = 140min$.

Finally, empty time can be reduced from $30min + 40min + 30min + 20min = 120min$ (either leaving or returning to the depot) to $30min + 10min + 10min + 20min = 70min$, thus producing a higher utilization of vehicle capacity, while serving all the demand in the same time as conventional vehicles.

Problem formulation

The problem formulation uses the nomenclature and parameter settings summarized in Table 1. The parameter values as stated in Table 1 are used in the experiments.

The MP-PDP is formulated as follows. Let $G = (N, A)$ be a directed graph where N is a set of nodes and A is a set of arcs. $ND = N_d \cup N_r \cup N_{sd}$ is the union of all depot nodes N_d , request nodes N_r , and service depot nodes N_{sd} . The set of depot nodes is defined as $N_d = N_d^+ \cup N_d^-$, where N_d^+ are the origin depot nodes and N_d^- are the destination depot nodes. For each depot, there is one pair of nodes from N_d in graph G . Similarly, the set of request nodes is defined as $N_r = N_r^+ \cup N_r^-$, with the pickup nodes $N_r^+ = N_{r,p}^+ \cup N_{r,f}^+$ for passenger and freight requests, respectively, and the drop off nodes $N_r^- = N_{r,p}^- \cup N_{r,f}^-$ for passenger and freight requests, respectively.

To simplify the mathematical problem formulation (compare Eqs. 1 - 49) each pair of depot nodes is duplicated by the maximum number of available platforms (κ). This ensures that at maximum all platforms can depart / arrive from / at the same physical depot but not at the same node in G . Similarly, the service depot nodes are duplicated by the maximum number of service depot visits (ϑ), which is an input parameter determining how often the same service depot can be visited by a platform. This formulation implies that a maximum number of ϑ modules per type must be available at each physical service depot location to facilitate feasible operations. With $h_r = |N_r|/2$ and $h_d = \kappa \cdot |N_d|/2$ as the number of requests and the number of depots, respectively, N can be defined as the set of nodes $N = \{1, \dots, h_d, h_d + 1, \dots, h_d + h_r, h_d + h_r + 1, \dots, h_d + 2 \cdot h_r, h_d + 2 \cdot h_r + 1, \dots, 2 \cdot h_d + 2 \cdot h_r + 2 \cdot h_r + 1, \dots, 2 \cdot h_d + 2 \cdot h_r + \vartheta \cdot n_{sd}\}$. The set of origin depot nodes (N_d^+) is $\{h_1, \dots, h_d\}$, the set of destination depot nodes (N_d^-) is $\{h_d + 2 \cdot h_r + 1, \dots, 2 \cdot h_d + 2 \cdot h_r\}$, the set of request pickup nodes (N_r^+) is $\{h_d + 1, \dots, h_d + h_r\}$, the set of request drop off nodes (N_r^-) is $\{h_d + h_r + 1, \dots, h_d + 2 \cdot h_r\}$, and the set of service depot nodes (N_{sd}) is $\{2 \cdot h_d + 2 \cdot h_r + 1, \dots, 2 \cdot h_d + 2 \cdot h_r + \vartheta \cdot n_{sd}\}$. The corresponding pickup/drop off

Table 1 Nomenclature and parameter values for the multi-purpose pickup and delivery problem

Notation	Description	Unit	Value
i, j	Node index	-	-
k	Platform index	-	-
m	Module index	-	-
u	Module usage index	-	-
l	Depot group index	-	-
$x_{i,j,k}$	Binary decision variable platform assignment	-	-
$y_{i,j,m}$	Binary decision variable module assignment	-	-
$s_{i,k}$	Continuous decision variable platform arrival time	-	-
$e_{k,u}$	Binary decision variable module usage	-	-
c_i	Continuous decision variable module capacity	-	-
N	Set of all nodes in the graph (requests, depots and service depots) $ND = N_r \cup N_d \cup N_{sd}$	-	-
N_r	Set of all request nodes $N = N^+ \cup N^-$	-	-
N_r^+	Set of all requests pickup nodes	-	-
N_r^-	Set of all requests drop off nodes	-	-
N_d^+	Set of all origin nodes	-	-
N_d^-	Set of all destination nodes	-	-
N_{sd}	Set of all service depot nodes	-	-
M	Set of all available modules $M = M_p \cup M_f$	-	-
M_p	Set of all available passenger modules	-	-
M_f	Set of all available freight modules	-	-
K	Set of all available platforms	-	-
G_l	Set of depots in depot group $l \in N_d^+$	-	-
a_i	Lower time window bound for node $i \in ND$	sec	-
b_i	Upper time window bound for node $i \in ND$	sec	-
$t_{i,j}$	Travel time including service time at node i from node i to j for $i, j \in ND$	sec	-
q_i	Demand for node $i \in ND$	-	-
$w_{i,j}$	Travel distance from node i to j for $i, j \in ND$	m	-
h_r	Number of requests	-	-
h_d	Number of depots	-	-
v	Travel speed	km/h	20
κ	Max. number of platforms	-	-
μ_p	Max. number of passenger modules	-	-
μ_f	Max. number of freight modules	-	-
η	Max. range per platform	km	100
ϑ	Max. service depot visits per depot	-	5
γ_p	Max. capacity for passenger modules	pas.	16
γ_f	Max. capacity for freight modules	items	16

Table 1 continued

Notation	Description	Unit	Value
ζ	Large positive Number	-	-
α_{tt}	Cost of total vehicle travel time	EUR/h	6.9
α_{ps}	Cost of a platform	EUR	313.67
α_{ms}	Cost of a module	EUR	156.84
α_{td}	Cost of travel distance	EUR/km	0.1
α_{mc}	Cost of module change	EUR	8.8
α_{ud}	Cost of one unserved request	EUR	470.52

pair for one request can be identified using $(i, i + h_r) \forall i \in N_r^+$, while the corresponding origin/destination pair for one depot is $(i, i + h_d + 2 \cdot h_r) \forall i \in N_d^+$.

The set of available modules (M) is defined as $M = M_p \cup M_f$, where M_p are passenger modules and M_f are freight modules, with $M_p = \{1, \dots, \mu_p\}$ and $M_f = \{\mu_p + 1, \dots, \mu_p + \mu_f\}$, respectively. The set of all available platforms (K) is defined as $K = \{1, \dots, \kappa\}$, with κ being the maximum number of platforms.

Each node $i \in N$ has a demand q_i , service duration o_i , and request-specific time window (a_i, b_i) . For depot nodes and service depot nodes the demand is set to zero. For pickup nodes, the demand represents the number of passengers or packages that should be picked up at that node. The sum of the demand for a given request pair is zero, i.e., the demand for drop off nodes is the negative value of the corresponding pickup node. The service duration for packages and passengers, and the respective time windows, are specified for each node individually.

The set of all arcs A is defined as $A = \{(i, j) \forall i, j \in N\}$ and hence spans a fully connected graph. The travel time $t_{i,j}$ is computed with $t_{i,j} = d_{i,j}/v + o_i \forall i, j \in N$, where v is the average travel speed in the network G . In our formulation, we assume that platforms drive with a constant and known speed.

A homogeneous fleet of maximum κ vehicle platforms is available, each using a(n) (exchangeable) module $m \in M$ with a type specific capacity. We assume that a fixed number of modules are available at each service depot at each point in time, defined by the parameter ϑ . Vehicles experience a service time at a service depot and a regular depot which represents the time needed for any loading and/or unloading. The distance travelled by any individual platform cannot exceed the maximum range value η .

The decision variables for the problem are $x_{i,j,k}$, $y_{i,j,m}$, $e_{k,u}$, $s_{i,k}$, and c_i . $x_{i,j,k}$ is a binary variable which is 1 if the node pair (i, j) is served by the platform k , and 0 otherwise. $y_{i,j,m}$ is a binary variable which is 1 if node pair (i, j) is served by module m , and 0 otherwise. $e_{k,u}$ is a binary variable which is 1 if platform k used modules u and 0 otherwise. This variable is used to keep track of the total usage of modules. $s_{i,k}$ is a positive real number that indicates the arrival time of vehicle platform k at node i . c_i is a positive real number which indicates the current load at node i .

The objective function parameters in the model are total vehicle travel time cost (α_{tt}), the fixed cost of a platform (α_{ps}), the fixed cost of a module (α_{ms}), the cost of travel distance (α_{td}), the fixed cost of module change (α_{mc}), the fixed cost of one unserved request (α_{ud}).

The objective function is composed of the minimization of six cost terms (see Eq. 1).

$$\min_{x,y,s,e,c} \alpha_{ps} \cdot B_1 + \alpha_{ms} \cdot B_2 + \alpha_{td} \cdot B_3 + \alpha_{tt} \cdot B_4 + \alpha_{mc} \cdot B_5 + \alpha_{ud} \cdot B_6 \quad (1)$$

The first term in the objective function computes the total number of platforms used in a solution (see Eq. 2). Note that index i is summed over the origin nodes N_d^+ . Since an origin node is visited only by one platform, the total sum equals the number of platforms.

$$B_1 = \sum_{i \in N_d^+} \sum_{j \in N_r} \sum_{k \in K} x_{i,j,k} \quad (2)$$

Equation 3 computes the total number of modules used utilizing the module usage decision variable e which is equal to 1 if u modules are used on platform k . The index u is also used as an integer number with $\{u \in \mathbb{Z} \mid 0 \leq u \leq |N|\}$.

$$B_2 = \sum_{u \in N} \sum_{k \in K} u \cdot e_{k,u} \quad (3)$$

Equation 4 computes the total distance traveled by all platforms in the solution.

$$B_3 = \sum_{i \in N} \sum_{j \in N} \sum_{k \in K} d_{i,j} \cdot x_{i,j,k} \quad (4)$$

The fourth term (see Eq. 5) computes the total vehicle travel time by subtracting the departure time from the origin depot from the arrival time of the corresponding destination depot. Note that for unused depots the difference equals zero, as is imposed by Eq. 44 in the problem formulation.

$$B_4 = \sum_{i \in N_d^+} \left(\sum_{k \in K} s_{i+2 \cdot h_r + h_d, k} - \sum_{k \in K} s_{i, k} \right) \quad (5)$$

Equation 6 computes the number of service depot visits, in combination with Eqs. 17 and 18 this equals the number of module changes.

$$B_5 = \sum_{i \in N} \sum_{j \in N_{sd}} \sum_{k \in K} x_{i,j,k} \quad (6)$$

The unserved demand is computed in Eq. 7 by subtracting the number of pickup nodes served from the total number of requests.

$$B_6 = h_r - \sum_{i \in N_r^+} \sum_{j \in N} \sum_{k \in K} x_{i,j,k} \quad (7)$$

In addition to the objective function, the following constraints constitute the proposed MP-PDP.

$$\sum_{j \in N} \sum_{k \in K} x_{i,j,k} \leq 1 \quad \forall i \in N, \quad (8)$$

$$\sum_{k \in K} x_{i,j,k} - \sum_{m \in M} y_{i,j,m} = 0 \quad \forall i \in N, j \in N, \quad (9)$$

$$\sum_{k \in K} x_{i,j,k} - \sum_{m \in M_p} y_{i,j,m} = 0 \quad \forall i \in N_{r,p}, j \in N, \quad (10)$$

$$\sum_{k \in K} x_{i,j,k} - \sum_{m \in M_f} y_{i,j,m} = 0 \quad \forall i \in N_{r,f}, j \in N, \quad (11)$$

$$\sum_{j \in N} x_{i,j,k} - \sum_{j \in N} x_{h_r+i,j,k} = 0 \quad \forall i \in N_r^+, k \in K, \quad (12)$$

$$\sum_{j \in N} y_{i,j,m} - \sum_{j \in N} y_{h_r+i,j,m} = 0 \quad \forall i \in N_r^+, m \in M, \quad (13)$$

$$\sum_{j \in N} x_{j,i,k} - \sum_{j \in N} x_{i,j,k} = 0 \quad \forall i \in N_r, k \in K, \quad (14)$$

$$\sum_{j \in N} y_{j,i,m} - \sum_{j \in N} y_{i,j,m} = 0 \quad \forall i \in N_r, m \in M, \quad (15)$$

$$\sum_{j \in N} x_{j,i,k} - \sum_{j \in N} x_{i,j,k} = 0 \quad \forall i \in N_{sd}, k \in K, \quad (16)$$

$$\sum_{j \in N} \sum_{m \in M_p} y_{j,i,m} - \sum_{j \in N} \sum_{m \in M_f} y_{i,j,m} = 0 \quad \forall i \in N_{sd}, \quad (17)$$

$$\sum_{j \in N} \sum_{m \in M_f} y_{j,i,m} - \sum_{j \in N} \sum_{m \in M_p} y_{i,j,m} = 0 \quad \forall i \in N_{sd}, \quad (18)$$

$$\sum_{j \in N} \sum_{k \in K} x_{i,j,k} \leq 1 \quad \forall i \in N_d^+, \quad (19)$$

$$\sum_{j \in N} x_{i,j,k} - \sum_{j \in N} x_{j,(i+2 \cdot h_r+h_d),k} = 0 \quad \forall i \in N_d^+, k \in K, \quad (20)$$

$$s_{i,k} + t_{i,j} - \zeta \cdot (1 - x_{i,j,k}) \leq s_{j,k} \quad \forall i \in N, j \in N, k \in K, \quad (21)$$

$$s_{i,k} + t_{i,h_r+i} - \zeta \cdot \left(1 - \sum_{j \in N} x_{i,j,k}\right) \leq s_{h_r+i,k} \quad \forall i \in N^+, k \in K, \quad (22)$$

$$s_{i,k} \geq a_i \quad \forall i \in N, k \in K, \quad (23)$$

$$s_{i,k} \leq b_i \quad \forall i \in N, k \in K, \quad (24)$$

$$\sum_{i \in N} \sum_{j \in N} w_{i,j} \cdot x_{i,j,k} \leq \eta \quad \forall k \in K, \quad (25)$$

$$c_i \leq \gamma_p \quad \forall i \in N_{r,p}, \quad (26)$$

$$c_i \leq \gamma_f \quad \forall i \in N_{r,f}, \quad (27)$$

$$c_j + \zeta \cdot (1 - y_{i,j,m}) \geq c_i + q_j \quad \forall i \in N, j \in N_r, m \in M, \quad (28)$$

$$c_i = 0 \quad \forall i \in (N_d^+ \cup N_{sd}), \quad (29)$$

$$\sum_{u \in N} u \cdot e_{k,u} - \sum_{i \in (N_d^+ \cup N_{sd})} x_{i,j,k} = 0 \quad \forall j \in N, k \in K, \quad (30)$$

$$\sum_{i \in G_l} \sum_{j \in N} x_{i,j,k} \leq 1 \quad \forall l \in N_d^+, k \in K, \quad (31)$$

$$x_{i,i,k} = 0 \quad \forall i \in N, k \in K, \quad (32)$$

$$y_{i,i,m} = 0 \quad \forall i \in N, m \in M, \quad (33)$$

$$\sum_{j \in N} \sum_{k \in K} x_{i,j,k} = 0 \quad \forall i \in N_d^-, \quad (34)$$

$$\sum_{j \in N} \sum_{m \in M} y_{i,j,m} = 0 \quad \forall i \in N_d^-, \quad (35)$$

$$x_{i,j,k} = 0 \quad \forall i \in N_r^+, j \in (N_{sd} \cup N_d^-), k \in K, \quad (36)$$

$$y_{i,j,m} = 0 \quad \forall i \in N_r^+, j \in (N_{sd} \cup N_d^-), m \in M, \quad (37)$$

$$x_{i,j,k} = 0 \quad \forall i \in (N_{sd} \cup N_d^+), j \in N_r^-, k \in K, \quad (38)$$

$$y_{i,j,m} = 0 \quad \forall i \in (N_{sd} \cup N_d^+), j \in N_r^-, m \in M, \quad (39)$$

$$x_{i,j,k} = 0 \quad \forall i \in N_{sd}, j \in N_{sd}, k \in K, \quad (40)$$

$$y_{i,j,m} = 0 \quad \forall i \in N_{sd}, j \in N_{sd}, m \in M, \quad (41)$$

$$x_{i,j,k} = 0 \quad \forall i \in N_d^+, j \in N_{sd}, k \in K, \quad (42)$$

$$y_{i,j,m} = 0 \quad \forall i \in N_{sd}, j \in N_d^-, m \in M, \quad (43)$$

$$s_{i,k} + \zeta \cdot \sum_{j \in N} x_{j,i,k} \geq b_{(i-2 \cdot h_r - h_d)} \quad \forall i \in N_d^-, k \in K, \quad (44)$$

$$x_{i,j,k} = 0 \text{ or } 1 \quad \forall i, j \in N, k \in K, \quad (45)$$

$$y_{i,j,m} = 0 \text{ or } 1 \quad \forall i, j \in N, m \in M, \quad (46)$$

$$e_{k,u} = 0 \text{ or } 1 \quad \forall u \in N, k \in K, \quad (47)$$

$$s_{i,k} \geq 0 \quad \forall i \in N, k \in K, \quad (48)$$

$$c_i \geq 0 \quad \forall i \in N \quad (49)$$

Equation 8 guarantees that each node is visited at most once. The inequality relation here and in Eq. 19 mean that not every node has to be visited, hence allowing for unserved demand. In Eq. 9 the platform assignment is connected with the module assignment to guarantee that each served node pair i, j is visited by a platform and a module.

In Eqs. 10 and 11 the request type and module type are required to match, that is, passenger requests have to be served with passenger modules and freight requests with freight modules. Constraints (12) and (13) guarantee that the pickup node and its corresponding delivery node are served by the same platform and module. Similarly, Eqs. 14 and 15 ensure that the same platform and module that enter a node also leave that node (i.e., flow conservation constraints).

In constraints (16)–(18) the handling of platforms and modules in service depots is described. Equation 16 assures that the same platform entering a service depot node also leaves that service depot, whereas the module and its module type have to change as formulated in Eqs. 17 and 18.

Each platform is required to start at an origin depot (compare Eq. 19) and end its trip at the corresponding destination depot (see Eq. 20). In Eq. 21 the arrival times for consecutively visited nodes are constrained. Note that the travel time $t_{i,j}$ between a node pair i, j includes the service time for node i . The large positive integer ζ is defined as $\max(b_i + t_{i,j} - a_j \forall i, j \in N)$.

In constraint (22) it is defined that the pickup node is served before the corresponding drop off node. Equations 23 and 24 guarantee that each arrival time is within its time window. In constraint (25) the length of each platform trip is constrained by the platform range.

The set of constraints (26)–(29) guarantee that each module type is only filled up to its type-specific capacity (compare Eqs. 26 and 27). Additionally, the capacity conservation is

assured by Eq. 28. Note that drop off nodes have the negative demand of its corresponding pickup node. In Eq. 29 the module loads are initialized to zero.

Using Eq. 30 the number of used modules can be computed. The number of origin depot node N_d^+ or service depot node N_{sd} visits by a platform k is equal to the number of modules used. Note that $\{u \in \mathbb{Z} | 0 \leq u \leq |N|\}$ is used as an index and integer variable. Due to duplication of depot nodes, the constraint (8) does not prevent the same platform from departing from the same physical depot twice. Therefore, additional constraints are required to ensure that the same platform k does not depart from the same depot twice; in Eq. 31 this is formalized. G_l contains a list of all duplicate nodes in the origin depot $l \in N_d^+$.

In Eqs. 32 - 44 several simplifications are formalized. Equations 32 and 33 prevent looping at any node. Equations 34 and 35 prevent any departure from any destination node for platforms and modules, respectively. Equations 36 and 37 prevent arriving at a service depot or destination node after visiting a pickup node for platforms and modules, respectively. Equations 38 and 39 prevent arriving at a drop off node after visiting a service depot or origin depot node for platforms and modules, respectively. Equations 40 and 41 prevent traveling from any service depot to any other service depot for platforms and modules, respectively. Equation 42 prevents traveling to any service depot after an origin depot node and Eq. 43 prevents traveling from any service depot to a destination depot node. In Eq. 44 the arrival times for unserved destination nodes are set to the upper time window bound of the corresponding origin depot node using ζ as a large positive number. By this, it is ensured that unserved depot nodes are not contributing to the trip duration term (compare Eq. 5).

In the remaining Eqs. 45–49 the domains of the decision variable for platform assignment $(x_{i,j,k})$, module assignment $(y_{i,j,m})$, module usage $(e_{k,u})$, arrival times (s_i) and capacity (c_i) are defined, respectively.

Adaptive large neighborhood search

Small and medium-sized instances can be solved using the optimization software CPLEX. For larger instances, a heuristic optimization algorithm is developed. We have adopted and implemented the ALNS algorithm, originally proposed by Ropke and Pisinger (2006), with problem-specific heuristics. The ALNS has been successfully applied to several PDPs (e.g., Sacramento et al. 2019; Masson et al. 2013) and has shown good performance for complex combinatorial problems in general. The basic idea of the ALNS algorithm is to iteratively destroy and repair solutions. In Algorithm 1 the general outline of the implemented ALNS is given.

In the first step, an initial feasible solution x is created. This can be done using one of the repair operators, as each of them results in a feasible solution. The operator weights are all initialized to 1. In the next step, one destroy and one repair operator are selected using the roulette wheel selection process. The probability $p_{i,j}$ of choosing operator $i \in O$, with O being the list of available operators, in iteration j is calculated using Eq. 50. Here, $w_{i,j}$ is the weight for operator i at iteration j . Note that there is a separate list (O) for destroy and repair operators, hence their probabilities are independent. The destroy and repair operators implemented are detailed below.

$$p_{i,j} = \frac{w_{i,j}}{\sum_{k \in O} w_{k,j}} \quad (50)$$

A candidate solution x' is created by sequentially applying both operators. Note that after both operators have been applied, the candidate solution is always feasible. In the last step, the

Algorithm 1 An outline of the ALNS framework (see Pisinger and Ropke 2007)

Data: passenger/freight demand, (service) depot positions, parameter settings

Result: best solution (x^*) for the MD-PDP

```

1 create a feasible solution ( $x$ ), set  $x^* := x$ ;
2 repeat
3   roulette wheel selection (see Eq. 50) for a destroy & a repair operator using weights;
4   create a destroyed solution ( $x_d$ ) using the chosen destroy operator on  $x$ ;
5   create a candidate solution ( $x'$ ) using the chosen repair operator on  $x_d$ ;
6   if  $Objective(x') < Objective(x^*)$  then
7     set  $x^* := x'$ ;
8     set  $x := x'$ ;
9     set score for chosen destroy & repair operator to  $\sigma_1$ ;
10  else
11    if  $x'$  is accepted (Simulated Annealing) then
12      set  $x := x'$ ;
13      if  $Objective(x') < Objective(x)$  then
14        set score for chosen destroy & repair operator to  $\sigma_2$ ;
15      else
16        set score for chosen destroy & repair operator to  $\sigma_3$ ;
17      end
18    else
19      set score for chosen destroy & repair operator to  $\sigma_4$ ;
20    end
21  end
22  update weights using new operator scores (see Eq. 51)
23 until maximum number of iterations, or objective variation threshold;
24 return  $x^*$ 

```

current solution x , global best solution x^* and the heuristic scores are updated according to the objective value of x' if the candidate solution x' is accepted. If a new overall best solution is found the candidate solution is set as the new global best solution and the new current solution, and the score for the operators is set to σ_1 . If the candidate solution is accepted using a simulated annealing decision process, the current solution is set to the candidate solution. If the objective value of the accepted candidate solution is lower than the objective value of the current solution the operator scores are set to σ_2 , else the score is set to σ_3 . If the candidate solution x' is not accepted, the best global solution and current solutions remain unchanged, and the operator scores are set to σ_4 . At the end of each iteration j the operator weights ($w_{i,j}$) for the chosen operators i are updated using Eq. 51. Here, the score ($s_{i,j}$) for the operator chosen i at the current iteration j is set to the corresponding σ value as outlined in Algorithm 1. The values of σ and the computation of the updated weights is in line with the process described in Ropke and Pisinger (2006).

$$w_{i,j+1} = w_{i,j} \cdot \delta + (1 - \delta) \cdot s_{i,j} \quad (51)$$

In Eq. 51 the operator decay parameter δ with $\{\delta \in \mathbb{R} \mid 0 \leq \delta \leq 1\}$ influences the speed at which the operator weights are adjusted to the scores. For low δ the adjustment rate is fast, while for high δ the adjustment rate is slow and the computed score values do not influence the operator's weights much. The algorithm terminates once the maximum number of iterations (λ) is reached or the change in the objective value - after a minimum number of iterations (λ_{min}) is performed - over an iteration span (ω) is below a threshold (ϵ). The termination

criterion is computed using Eq. 52 at each iteration i .

$$\frac{\sum_{j=i-2\omega}^{i-\omega} z_j}{\sum_{j=i-\omega}^i z_j} - 1 \leq \epsilon \quad \text{for } \lambda_{\min} \leq i \leq \lambda, \quad (52)$$

with z_j being the objective value at each iteration.

Heuristic operators

The implemented operators are in line with the conventionally used operators of Ropke and Pisinger (2006) and Sacramento et al. (2019). The interested reader is referred to these publications for more detailed information. In Hatzenbühler et al. (2023) the authors propose heuristic operators to solve a PDP with modular vehicles. In their study the vehicles can form platoons but are limited to transporting a single demand type. The operators in this work are adjusted to accommodate the novel multi-purpose modules and vehicle operations, hence resulting in novel destroy and repair heuristics. In the following paragraphs only a brief description of each implemented operator and adjustments for the special nature of the MP-PDP are given.

Destroy Operators The general idea of the destroy operators is to help diversify the search process and thereby explore the solution space. In the implemented destroy heuristics depots (origin and corresponding destination node), service depots or requests (a pair of pickup and drop off nodes) are removed from a solution. Hence, except for service depots, no single nodes are removed from the solution but rather a node pair.

- **Random removal:** In this heuristic, a random selection of currently served requests (pickup and drop off node) are removed from the solution and considered as unserved.
- **Module removal:** In this heuristic a random selection of currently used modules is made. All requests served by any of the chosen modules, including potential service depot nodes, are removed from the solution and considered as unserved.
- **Platform removal:** In this heuristic, one of the currently used platforms is chosen at random. All requests served by this platform, including potential service depot nodes, are removed from the solution and considered as unserved.
- **Service depot removal:** In this heuristic a random selection of currently used service depot nodes are removed from the solution. A service depot node can only be removed if that removal results in a feasible solution. Hence, in this heuristic redundant service depots such as two consecutive service depots, or service depot before/after a depot are removed. The removal of service depots including the removal of associated platforms is achieved with the module removal and platform removal heuristics.
- **Shaw removal:** This removal heuristic was first proposed by Shaw (1997) and removes similar requests from the solution. Here, the relatedness is computed using Eq. 53, using a distance term, travel time term, and request load term. To diversify the removal process, a determinism parameter is introduced; see detailed description in Ropke and Pisinger (2006).
- **Worst removal:** In this heuristic, the requests with the highest cost contribution to the total cost are removed. The cost contribution for each served request is calculated by computing the objective function with and without this request.

The relatedness ($R_{i,j}$) for two requests i and j , as utilized in the *Shaw removal* operator is computed with

$$R_{i,j} = \phi \cdot (w_{a_i,a_j} + w_{b_i,b_j}) + \chi \cdot (|s_{a_i} - s_{a_j}| + |s_{b_i} - s_{b_j}|) + \psi \cdot (|q_i - q_j|), \quad (53)$$

where ϕ , χ , and ψ are input parameters for the distance term, travel time term and the load term, respectively. a_i and b_i are the pickup and drop off nodes for request i . $w_{i,j}$, t_i , and q_i are the travel distance, arrival time, and demand for requests i , j , respectively.

The handling of service depot nodes introduces a few additional considerations that are novel to the proposed problem. In all destroy operators except for *Module removal* and *Platform removal*, a service depot node is only removed from the solution if that temporary solution remains feasible. Therefore, a removal would be feasible if a service depot is at the beginning or at the end of the route, or if two service depots are visited consecutively. In the *Module removal* and *Platform removal* operators, either an entire trip or a trip segment (i.e. depot to service depot, service depot to service depot, service depot to depot, or depot to depot) is removed. These operators strongly diversify the solution and guarantee that the number of platforms and modules is optimized. Additionally, these heuristics remove service depots from a solution, which is essential to further explore the solution space.

In all destroy operators except for the *Platform removal*, the number of requests or modules that are removed is a random integer (N) following the discrete uniform distribution over the set

$$N = \{\iota, \dots, \min(n_{served}, n_r \cdot \xi)\}, \quad (54)$$

with ι and ξ being input parameters determining how many requests are removed. For high values of ξ the maximum number of removed requests equals the number of served requests (n_{served}), while for low values of ξ only a fraction of the total number of request (n_r) in the scenario are removed.

Repair Operators As a first step for all repair operators, the list of unserved requests is randomized. By this step, inserting the same requests in the same order multiple times can be avoided. Next, we loop through the list of requests and try to insert each request individually following the principles described below. Additionally, we insert individual nodes (service depots) and node pairs (depots and requests) only in feasible positions in the temporary solution. This results in (1) at the end of each operator, we guarantee a feasible solution, and (2) certain solutions cannot be created immediately. An example of the second consequence is the situation when a request can only be inserted if a service depot is also added. In the current implementation of the algorithm, these cases are handled through the high number of iterations performed and the combination of intensification and diversification heuristics.

At the end of each operator redundant service depots are removed. This applies in the following three cases: (1) a service depot is at the beginning of a route, (2) a service depot is at the end of a route, or (3) two service depots are visited consecutively.

- First fit insert: In this repair heuristic, all unserved requests, depot nodes and service depot nodes are inserted into the first feasible location.
- Inter route insert: In this repair heuristic all unserved requests, depot nodes and service depot nodes are inserted into the best feasible location of *one* route. This route equals the route from which this request/node was previously removed. If the request/node has never been served, a random route is chosen.
- Best insert: In this repair heuristic all unserved requests, depot nodes and service depot nodes are inserted into the best feasible location of *all* routes.

For *Inter route insert* and *Best insert* the best insertion position for each request/node is found using a two-step approach. In a first step, all feasible positions within a request route are computed. Then for each position, compute the objective value of the new route with the request/node inserted. The position with the smallest delta value is chosen as the final insertion position.

The re-/insertion of service depots in any of the three repair operators follows the following three principles. These principles are applicable to any route. (1) A service depot node can be inserted as the first node visited after the depot, (2) a service depot node can be inserted as the last node before ending the route, or (3) a service depot node can be inserted between any node pair, if the first node of that pair is a drop off node and the second node of that pair is a pickup node, and both these nodes are of different request type. The reconfiguration of platforms is achieved, since the list of requests and unserved (service) depots is randomized before each repair operator, and hence service depots can be inserted early or late in the repair process resulting in various new route configurations.

Experimental design

To analyze the capabilities of the multi-purpose vehicle, several experiments are performed using the model described in the previous section.

Scenario definitions

The scenarios are created by varying the spatial demand distribution, time window constraints, and number of service depots. In total 54 scenarios represent different vehicle use cases and operations. In the following paragraphs, the different configurations are described. The analysis focuses on three main dimensions influencing the operation of the new vehicle concept: the temporal and spatial distributions of the demand, and the number of available service depots. Each scenario is solved using the parameters from Table 1. Scenarios without service depots model the conventional vehicle operations and function as the base case in the following analysis. The base demand for passengers and freight is in line with real data from the area of Stockholm, Sweden. For each scenario an ensemble run of 10 independent optimizations is performed, the average values of the ensemble runs are used for the analysis.

Spatial demand distribution

Figures 3a–c represent the different spatial demand distributions analyzed in this paper. For all scenarios the demand consists of 50 passengers and 50 freight requests. The locations of the depots (red) correspond to those of freight distribution centers in Stockholm. The locations of service depots (blue) are chosen at strategic points in the service area to best illustrate the transport concept. The locations of depots and service depots remain unchanged among the different scenarios.

In scenarios 1–18 as described in Table 3 (see Fig. 3a) the passenger requests (pickup and drop off locations) are evenly distributed over the entire service area. The freight requests are focused on the central area in Stockholm. This central demand pattern represents urban deliveries operations that selectively serve, e.g., stores, shops and grocery stores only in a centralized area.

The spatial distribution of scenarios 19–36 from Table 3 (see Fig. 3b) represents city-wide deliveries spread evenly over the entire area of interest and typical passenger movement patterns throughout the city. The passenger pickup and drop off points are different to Fig. 3a but share the same generation process.

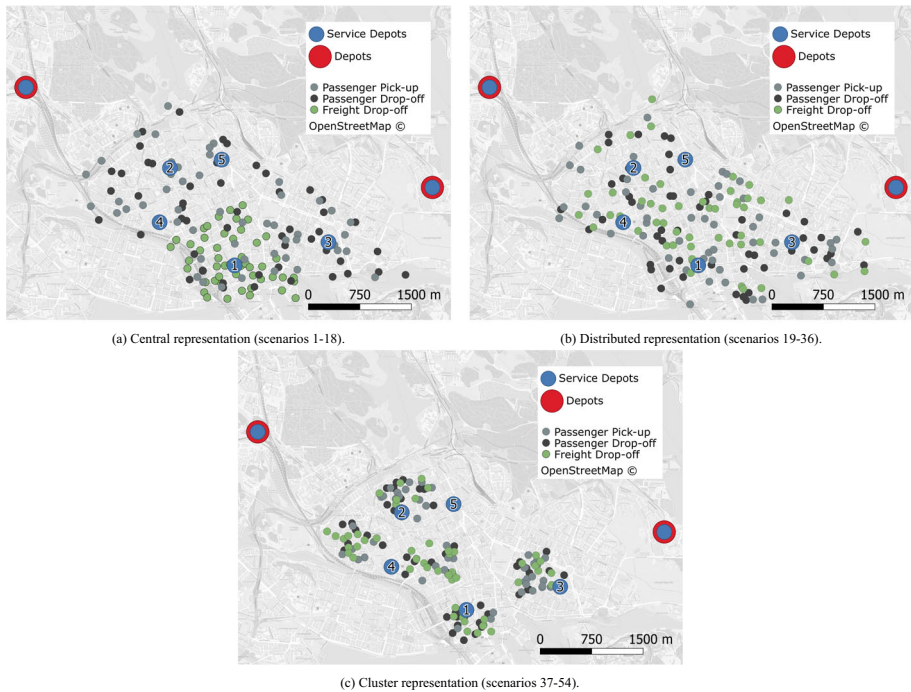


Fig. 3 Spatial demand representation for the three different configurations

In the third variation in demand distribution (Fig. 3c and scenarios 37–54 in Table 3), the passenger and freight requests form spatial clusters. A request can however be transported between clusters, i.e., pickup and drop off locations do not need to be in the same cluster. An operator serving only a certain area of a city or having dedicated areas of interest would face a similar demand distribution as is represented in this scenario.

Time window settings

Next to the spatial demand variation, each scenario also has different time window definitions (compare Table 2). In total three different time window settings are implemented. The first setting represents a scenario where no time window constraints are present so that each request can be served at any time. This setting is chosen to represent a base scenario for the optimization. Due to the absence of time windows the solution space is significantly increased and therefore finding a robust solution is more challenging for the algorithm. At the same time creating feasible solutions is simplified and the general functionality of the algorithm can be better shown.

The second time window setting represents loose peak time window constraints, meaning that each request has a dedicated time for pickup and drop off but large deviations are permissible, e.g., several hours. The distribution of the requests follows a peak distribution, where passenger demand is highest during the morning and afternoon hours and freight demand is highest during mid-day.

In the last time window setting, the constraints are tightly set. All requests have an arrival time window of 20 min, while the departure time window is set to be nonrestrictive. Similar

Table 2 Time window definitions

Start Time [sec]	End Time [sec]	Type		Count
No time windows				
1	86400	drop off	freight	50
1	86400	drop off	passenger	50
1	86400	pickup	freight	50
1	86400	pickup	passenger	50
Peak time windows				
28800	43200	drop off	passenger	25
28800	43200	pickup	passenger	25
36000	57600	drop off	freight	50
36000	57600	pickup	freight	50
50400	64800	drop off	passenger	25
50400	64800	pickup	passenger	25
Tight time windows				
1	86400	pickup	freight	50
1	86400	pickup	passenger	50
24600	25800	drop off	passenger	2
25800	27000	drop off	passenger	3
27000	28200	drop off	passenger	5
28200	29400	drop off	passenger	6
29400	30600	drop off	passenger	5
30600	31800	drop off	passenger	3
31800	33000	drop off	passenger	1
33000	34200	drop off	passenger	1
34200	35400	drop off	freight	1
35400	36600	drop off	freight	1
36600	37800	drop off	freight	1
37800	39000	drop off	freight	1
39000	40200	drop off	freight	2
40200	41400	drop off	freight	4
41400	42600	drop off	freight	4
42600	43800	drop off	freight	5
43800	45000	drop off	freight	6
45000	46200	drop off	freight	6
46200	47400	drop off	freight	5
47400	48600	drop off	freight	4
48600	49800	drop off	freight	4
49800	51000	drop off	freight	2
51000	52200	drop off	freight	1
52200	53400	drop off	freight	1
53400	54600	drop off	freight	1
54600	55800	drop off	freight	1

Table 2 continued

Start Time [sec]	End Time [sec]	Type	Count
55800	57000	drop off passenger	1
57000	58200	drop off passenger	2
58200	59400	drop off passenger	3
59400	60600	drop off passenger	5
60600	61800	drop off passenger	6
61800	63000	drop off passenger	5
63000	64200	drop off passenger	2

to the second setting, the requests follow a peak distribution, i.e., passenger requests peak during the morning and afternoon and freight request peak during mid-day.

Variation in available service depots

The third dimension of the scenario definition is the number of available service depots. Each spatial and temporal configuration is combined with six different service depot variations, increasing from no available service depot to five available service depots in the city center (compare Fig. 3). In all scenarios the same service depot is added first, second, third, etc. The location and sequence of the service depots are based on practical considerations, i.e. the first depot at the most central location and the other depots spread out evenly over the city. In the scenarios with at least one service depot the regular depots can also be used as service depots. The scenarios without available service depots represent the operations with conventional vehicles.

In Table 3 an overview of the configurations of all 54 scenarios is given.

Parameter settings

The parameters used for the experiments are determined based on parameter tuning procedures, reported literature values or general reasoning. If not stated otherwise the parameters in Tables 1 and 4 are used for the remainder of this paper.

Table 3 Scenario overview

Scenario	Spatial distribution	Time window variation	Nb. of service depots
1–6	Central	noTW	(0, 1, 2, 3, 4, 5)
7–12	Central	peak	(0, 1, 2, 3, 4, 5)
13–18	Central	tight	(0, 1, 2, 3, 4, 5)
19–24	Distributed	noTW	(0, 1, 2, 3, 4, 5)
25–30	Distributed	peak	(0, 1, 2, 3, 4, 5)
31–36	Distributed	tight	(0, 1, 2, 3, 4, 5)
37–42	Clustered	noTW	(0, 1, 2, 3, 4, 5)
43–48	Clustered	peak	(0, 1, 2, 3, 4, 5)
49–54	Clustered	tight	(0, 1, 2, 3, 4, 5)

Table 4 ALNS parameter

Notation	Description	Value
σ_1	Operator weight for new best overall solution	7
σ_2	Operator weight for better candidate solution	2
σ_3	Operator weight for accepted solution	9
σ_4	Operator weight for rejected solution	1
δ	Operator decay per iteration	0.1
λ	Iterations	10000
λ_{min}	Min. iterations	2000
ω	Iterations look-back	1000
ϵ	Objective improvement threshold	0.01
T_{start}	Start temperature simulated annealing	100
T_{end}	End temperature simulated annealing	0.0001
ν	Step size simulated annealing	0.9999
ϕ	Relatedness parameter for distance	9
χ	Relatedness parameter for time	3
ψ	Relatedness parameter for load	2
ρ	Randomising selection parameter for shaw removal requests	6
ρ_{worst}	Randomising selection parameter for worst removal requests	3
ξ	Max. request removal factor	0.32
ι	Min. request removals	1

Model parameters

The platform and operational parameters, i.e., maximum trip range and passenger/freight capacity are based on vehicle design parameters published by manufacturers or project reports. The travel time costs are based on a value-of-time of 6.9EUR/h as mentioned in Börjesson and Eliasson (2014).

Militão and Tirachini (2021) present a linear estimation model based on vehicle capacity to estimate the distance- and time-based operational costs for electric vehicles. The model is built using data from Munich, Germany, and Santiago, Chile. For the distance related cost their model is $\alpha_{td} = 0.003599 \cdot \gamma_k + 0.04162$. For the time related costs their model equals $\alpha_{ps} = (0.1753 \cdot \gamma_k + 16.8) \cdot \frac{2}{3} \cdot 24$, and $\alpha_{ms} = (0.1753 \cdot \gamma_k + 16.8) \cdot \frac{1}{3} \cdot 24$, respectively, using the capacity γ_k per type $k \in K$. The parameters α_{ps} and α_{ms} are scaled with $2/3$ and $1/3$, respectively, to account for cheaper module operation costs. The second factor, 24, represents the duration of the planning period in hours.

The module change cost is estimated based on an average salary of 17.6EUR/h (Economic Research Institute 2024) for a bus driver in Sweden and the assumption that one person would need approximately 30 min to switch the module.

The parameter settings for unserved demand are chosen to match the fleet size related costs for one platform and one module. This is to avoid a single vehicle serving a single request.

There are no differences in model parameters between conventional operations and multi-purpose operations. This is done to isolate the impact of the new vehicle technology on the results.

In preparation for the final results, two experiments are performed. Both have the goal to overcome the challenges imposed by the numerical differences in model parameters and to achieve robust optimization of the vehicle fleet and vehicle routes. The first experiment uses a hierarchical objective function. The top-level problem optimizes the number of platforms and modules, while the low-level problem optimizes the vehicles routes using the results from the top-level problem. The objective function in this approach is adjusted to only represent the relevant cost terms for each subproblem. In the second experiment, the vehicle fleet and routes are optimized simultaneously using all cost terms in the objective function. When comparing the results of both experiments for all scenarios, no significant differences in the solutions could be seen. Hence, the results reported in “[Results](#)” are achieved using the simultaneous optimization approach of the second experiment.

Additionally, we employ a two-step approach. First, the scenarios with conventional vehicle operations are solved using the ALNS as described in “[Methodology](#)”. Second, the best solutions of the conventional operations are used as initial solutions for the scenarios with multi-purpose vehicle operations. Since all solutions with conventional operations are also feasible for the multi-purpose scenarios this improves the optimization process for the multi-purpose vehicles by initiating their optimization with high quality solutions.

Optimization parameter

Following the principle of Ropke and Pisinger (2006) an iterative grid search parameter tuning approach has been performed. For each parameter in Table 4 a discrete set of 10 possible parameter values each has been manually specified. Additionally, 36 scenarios have been created synthetically with varying spatial/temporal demand distribution and number of (service) depots to have a diverse set of scenarios. Each scenario has a total of 40 requests, which represents a compromise between fast computation times and scenario complexity. The final parameter values are reported in Table 4. These parameters correspond to the parameter values with the lowest average optimality gap over all scenarios compared to the best available solution.

Results

The result section is divided into two parts. In the first part, the ALNS and CPLEX solutions are compared for small and medium-sized problems to showcase the functionality and benchmark the ALNS algorithm. The second part presents and discusses the results obtained for large scenarios, focusing on the changes induced by the multi-purpose vehicle technology.

Validation of ALNS

The small and medium-sized problems are created synthetically with varied number of requests, number of depots, and varied time window definitions. As shown in Table 5, this variation leads to several problems for each value of the number of requests. In total 118 problems with varied complexity are created.

Table 5 Comparison of CPLEX and ALNS

Nb. of requests	Nb. of problems	Avg. Obj. deviation	Avg. time CPLEX [sec]	Avg. time ALNS [sec]
4	12	-4.11E-06	0.21	2.52
8	12	-2.13E-09	0.48	2.81
12	12	-2.65E-06	1.26	2.63
16	12	8.03E-10	2.62	3.11
20	12	-7.60E-07	11.07	4.43
24	12	3.08E-08	44.83	4.35
28	10	2.94E-08	422.70	5.04
30	24	6.62E-03	427.90	5.16
40	12	4.38E-08	1707.35	6.16

To validate the ALNS performance, each problem is solved using CPLEX and ALNS. For the ALNS approach, each problem is solved with an ensemble run of 10. The numerical values reported for the ALNS for each request group are based on the average values of each ensemble run. The maximum computation time for CPLEX is set to 1 h and the accepted optimization gap is set to 0.4%.

As can be seen in Table 5, the ALNS algorithm is capable of finding the optimal solution for all problems. The objective deviation is computed as the percentage difference between the best CPLEX solution and the average of all ensemble ALNS solution for a problem. The reported value in Table 5 is the objective deviation averaged over all problems with the same number of requests. With the CPLEX termination conditions six of the problems could not be solved to optimality within the permitted time limit, meaning that the ALNS is in all cases capable to match the performance of CPLEX. In fact, in 116 problems the ALNS always finds the exact optimal solution or better solutions than CPLEX due to exceeded time limit.

Looking at the average computation time for CPLEX and ALNS a clear trend can be seen. For larger problems starting from approximately 20 requests the ALNS computation time is shorter than that of CPLEX. The reported computation time for ALNS is the time until the algorithm terminates, e.g., over 2000 iterations. However, for most problems, the best solution is found after only a couple of hundred iterations, highlighting the rapid convergence of ALNS. It should be noted that the ALNS computation time is not linear with the number of requests.

Scenario analysis

In this section scenario-specific results with a focus on the spatial, temporal, and service depot variation are discussed. The average computation time for all scenarios is 1020 sec. The final best solution is found on average after 1200 iterations, while on average 2020 iterations are computed for each scenario.

Impact of spatial distribution and time window constraints

The figures in this section show aggregated values. The numerical values are average values for each aggregation. In Fig. 4 the objective value for the different definitions of the time

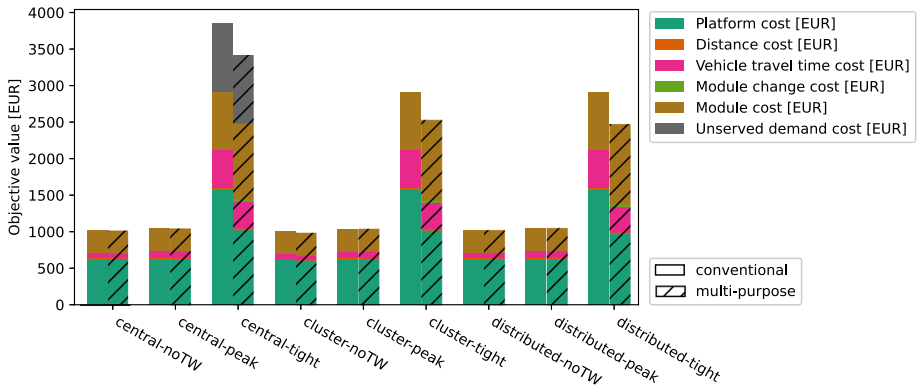


Fig. 4 Average objective values for different spatial distributions and time window definitions

window and the spatial groups is shown. The bars without pattern correspond to scenarios without service depots and therefore represent conventional operations, while the multi-purpose operations have between 1 and 5 additional service depots and are shown as bars with pattern. The different colors indicate the value of each cost term in the objective function.

It can be seen that for tight time window definitions the objective values are significantly higher than for peak and no time window definitions. This is because more platforms and more modules are needed to serve all requests. In addition, there is no significant difference in the objective value between operations with no time window and peak time window definitions. The objective value reduction are 11.3%, 13.4%, and 15.4% for central, cluster and distributed scenarios, respectively. Thus, multi-purpose operations are most beneficial for clustered and distributed scenarios.

The reduction for tight time window definition mainly stems from a reduction of used platforms of approximately 35%, which outweighs the higher costs for the modules required. In the conventional case additional vehicles have to be used to serve requests that cannot be served within the tight-time window, whereas the flexibility of the multi-purpose vehicles allows for a strategic change of the module so that more requests can be served with the same platform. In peak time window scenarios this advantage is not available since most requests can be served in an optimal way with the same amount of platforms. This principle can clearly be seen in Figs. 5 and 6. Additionally, the total vehicle travel time costs and distance costs are reduced by approximately 33% and 16%, respectively, when deploying multi-purpose vehicles. The significantly higher objective values for centralized scenarios with tight time windows compared to other tight time window scenarios can be explained with the unserved demand. In all these scenarios, 2 requests could not be served. This is independent of the vehicle technology used.

When analyzing the total distance traveled by all platforms in a solution (see Fig. 7) and the total travel time cost (see Fig. 8), the following general observations can be made. First, for peak and no time window scenarios the total distance and trip duration are significantly lower than for tight time window scenarios. This is as expected since the order in which requests can be served is less constrained. Hence the ALNS algorithm can generate more direct and therefore shorter platform routes, which in turn also results in shorter trip duration. This observation is independent of the vehicle technology.

Interestingly, for tight time window scenarios, the trip durations for multi-purpose vehicle operations are significantly shorter compared to conventional operations. This can be

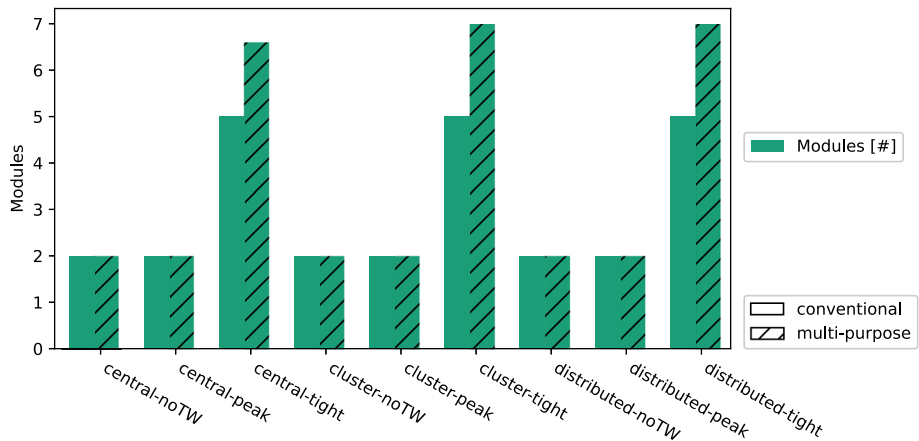


Fig. 5 Average number of modules utilized for different spatial distributions and time window definitions

explained by the fact that the multi-purpose vehicles utilize the centrally located service depots to start a new trip, which directly translates to shorter trip times. This also results in a reduction in the distance traveled (see Fig. 7). The explanation for this is two-fold. First, to achieve shorter trip duration and trip distances the order in which requests are served is different between conventional and multi-purpose operations. Second, due to the removal of additional trips to and from the main depots shorter travel distances and trip times can be recorded.

A second observation can be made by comparing the spatial distributions. For clustered scenarios without windows and with peak time windows, the total distance traveled by the platforms is significantly shorter than for the central and distributed scenarios. This indicates that short routes can be found when the demand is spatially clustered. The effect is however not present for the tight time window scenarios where modules are utilized (compare Fig. 5). This shows how the utilization of service depots and multi-purpose vehicles changes the order in which requests are served and where vehicles are operating.

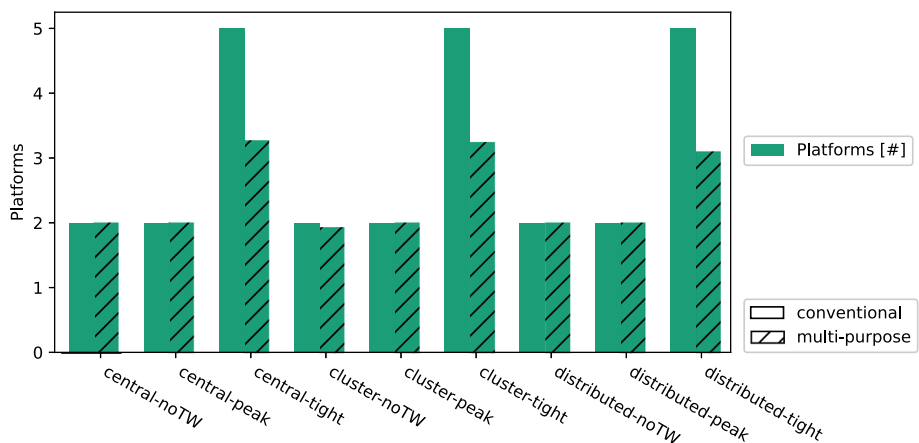


Fig. 6 Average number of platforms utilized for different spatial distributions and time window definitions

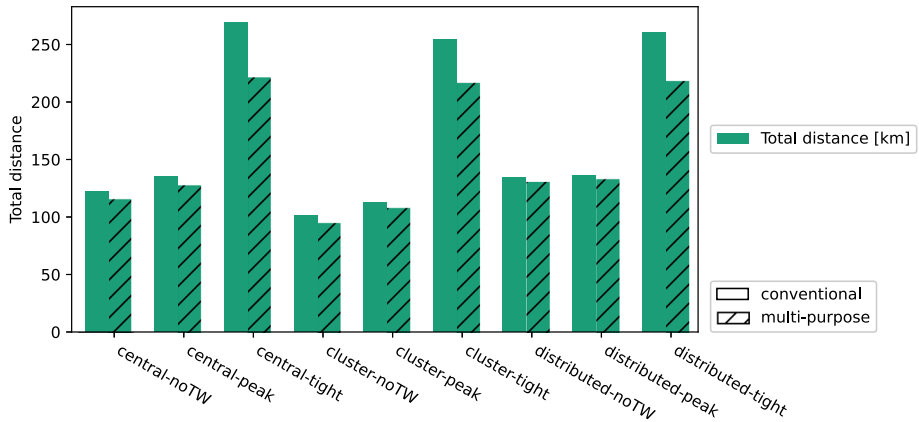


Fig. 7 Total distance traveled by all platform in a solution averaged for different spatial distributions and time window definitions

Impacts of module usage

As described in the previous section, the number of platforms needed to serve the demand can be reduced by utilizing multi-purpose vehicles. In Fig. 9 the number of platforms in a solution is shown with respect to the number of available service depots. Two main observations can be made from this figure. First, the visit of just one service depot, i.e., one module change, results in a reduction of approximately two platforms for all three spatial variations (central, cluster, and distributed). This is only true for scenarios with tight time window definitions. For other time window definitions, no or only small reduction in platforms due to visits at the service depot is observed. Second, no additional platforms are saved by having more service depots available. This indicates that there is an optimal combination of platforms and the number of modules, which cannot be improved by adding more service depots to the scenario.

This observation is confirmed by the average number of module changes per platform shown in Fig. 10. The different patterns in Fig. 10 indicate the different number of service

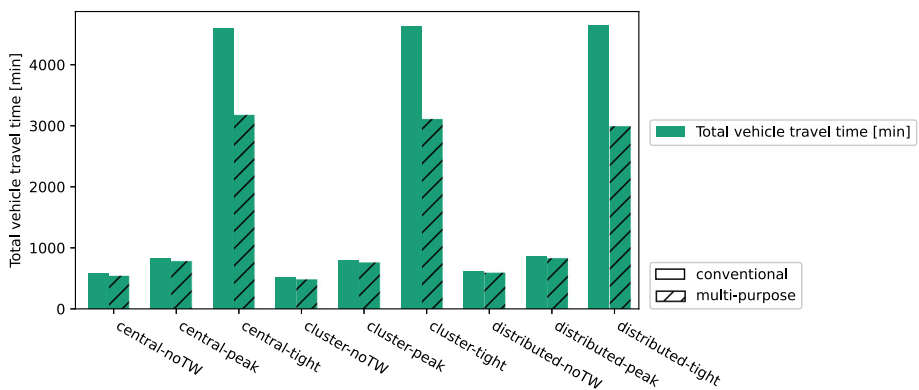


Fig. 8 Trip duration by all platforms in a solution averaged for different spatial distributions and time window definitions

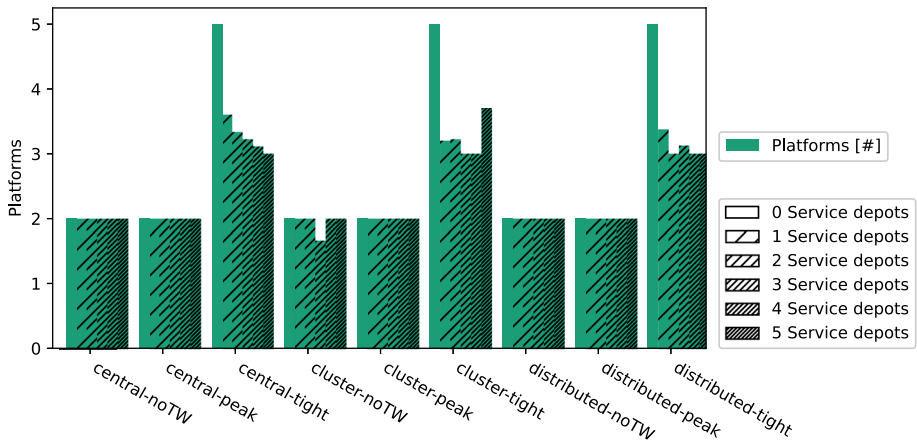


Fig. 9 Average number of platforms in a solution for different number of service depots, spatial distributions and time window definitions

depots in each scenario (compare Fig. 3). Note that the number of service depots indicates the order in which they are added. This order is always the same. The first service depot added is always the service depot at location 1, the second is always the service depot at location 2, and so on until all 5 service depots are added to the scenario.

As described in the previous section, the results for scenarios with peak time windows and without time window constraints exhibit similar trends. Most of the platforms in these scenarios do not change their purpose along their route, meaning that not every platform in these scenarios visits a service depot along its route. This is different with tight time window definitions, where an average of 1.18 module changes are made per platform. This means that few platforms visit multiple service depots during their trip. It can also be seen that the number of module changes does not correlate with the number of available service depots, i.e., for an increasing number of available service depots, the number of module changes does not increase significantly. This suggests that several service depots are rarely used and shows the importance of positioning the service depots at strategic locations within the service area.

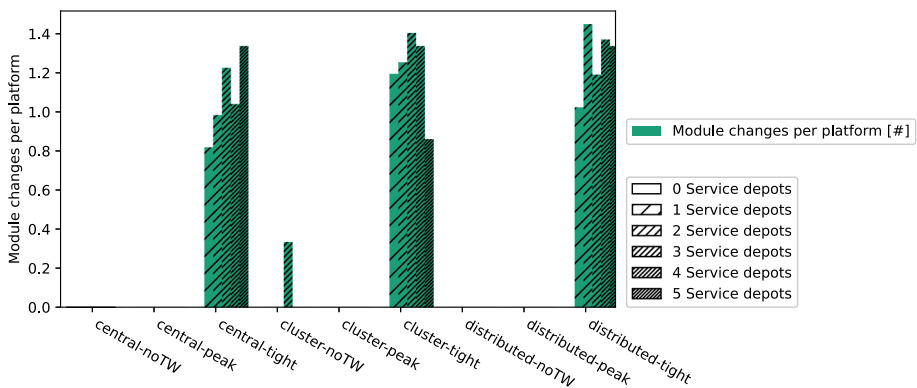


Fig. 10 Module changes per platform for different number of service depots, averaged for different spatial distributions and time window definitions

Discussion and conclusion

In this study, a new vehicle technology for sequential consolidation of urban passenger and freight flows is examined. By exchanging a removable module, different types of requests (passenger or freight) can be served using the same vehicle platform. The paper proposes an extension of the pickup and delivery problem to model these modular vehicle operations. We demonstrate the ability of an adjusted ALNS optimization algorithm to solve the routing problem for large-scale scenarios and highlight the benefits of modular vehicles over conventional operations.

The conducted experiments are based on scenarios in Stockholm, Sweden and show that the use of multi-purpose vehicles lead to an objective value reduction of 11.3%, 13.4%, and 15.4%, for scenarios with tight time windows and central, cluster and demand distribution, respectively. The reduction mainly stems from an approximate 35% reduction in the number of platforms required to serve requests in these scenarios. The reduction comes at the cost of approximately 37% more used modules. Furthermore, the results show that the availability of service depots leads to a different order in which the requests are served, resulting in an average reduction in the total duration of the vehicle trip of approximately 33% and a reduction in travel distance of approximately 16%. The number of unserved requests remains unchanged when changing the vehicle technology, indicating an unchanged level of service.

Considering the usage of service depots, the average number of module changes is around 1.18 for scenarios with tight time window definitions, showing that every platform is visiting at least one service depot along its route. The number of platforms saved does not correlate with the number of service depot available, meaning that more available service depots and/or more service depot visits do not lead to further reduction in required platforms. The main benefits can be achieved with one module change. Lastly, it can be noted that the number of module changes is not significantly increased by increasing the number of available service depots in the scenario. This indicates that most service depots are rarely used, and hence the positioning of service depots is crucial for the benefits of the proposed vehicle technology.

The results of this study can be used by practitioners and policymakers to conclude whether the combination of passenger and freight demand flows will yield benefits compared to existing systems. In general, it can be said that for realistic (tight) time window definitions the new vehicle technology leads to promising results, mainly a reduction in required platforms to serve the same demand. Generally, in comparison with simultaneous integration concepts the proposed sequential integration concept guarantees that passenger requests are not affected by freight delivery processes. Hence, the level of service for passengers is higher in this form of integration.

Our study is subject to several limitations affecting the scalability of the model and the generality of the results reported. The computation time of the ALNS is capable of solving problems of around 200–400 requests within a reasonable time, however, for larger problems, the proposed heuristic algorithm needs to be improved with, e.g., pre-processing steps (e.g., demand clustering), or multi-level optimization approaches. Further research may examine the sensitivity of numerical solutions to vehicle characteristics such as speed, range, and capacity. Additionally, the location and costs structures of depots and vehicle operations influence the generality of the proposed results. Furthermore, we assume that during every service depot visit the correct module type is always available. In the model this is guaranteed by setting the number of available modules per type and per service depot equal to the number of maximum allowed service depot visits, as defined by parameter ϑ . In this way, we can guarantee successful operation of the transportation system, but we overestimate the module

availability compared to a more realistic scenario. The current version of the proposed model does not guarantee that the number of modules and their types is the same before and after the planning period at each depot. In practical operations this could require re-balancing of the vehicles, which would reduce the benefits of the proposed multi-purpose transport system. Notwithstanding, the proposed modeling framework and optimization algorithm can be adapted to accommodate such considerations if necessary.

The proposed research can be extended in three directions. First, by integrating mixed-fleet properties (i.e., different vehicle sizes, simultaneous optimization of conventional and multi-purpose vehicles) for urban freight transport studies. Second, module scheduling and/or module inventory features can be added to the proposed formulation. Third, the (service) depot positioning problem can be integrated in the proposed model in order to expand to strategic planning decisions.

Acknowledgements This work was in part supported by the Sustainable and Integrated Transport Systems (Hållbara och Integrerade Transport System HITS) project under grant number 2020-00565 (Vinnova). The computations were enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC) at HPC2N partially funded by the Swedish Research Council through grant agreement no. 2018-05973.

Author Contributions CRediT author statement: Jonas Hatzenbühler: Formal analysis, Writing- Reviewing and Editing, Conceptualization, Methodology, Data curation, Writing original draft. Győző Gidófalvi: Formal analysis, Conceptualization, Methodology, Supervision, Project administration, Writing review and editing. Erik Jenelius: Formal analysis, Conceptualization, Methodology, Supervision, Project administration, Writing review and editing. Oded Cats: Formal analysis, Conceptualization, Methodology, Supervision, Writing review and editing.

Funding Open access funding provided by Royal Institute of Technology.

Declarations

Competing interests The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Alizadeh Foroutan, R., Rezaeian, J., Mahdavi, I.: Green vehicle routing and scheduling problem with heterogeneous fleet including reverse logistics in the form of collecting returned goods. *Appl. Soft Comput.* **94**, 106462 (2020). <https://doi.org/10.1016/j.asoc.2020.106462>
- Arslan, A., Agatz, N., Kroon, L., Zuidwijk, R.: Crowdsourced Delivery: A Dynamic Pickup and Delivery Problem with Ad-Hoc Drivers. SSRN Scholarly Paper ID 2726731. Social Science Research Network. Rochester, NY (2016). <https://doi.org/10.2139/ssrn.2726731>
- Beirigo, B.A., Schulte, F., Negenborn, R.R.: Integrating people and freight transportation using shared autonomous vehicles with compartments. *IFAC-PapersOnLine* **51**, 392–397 (2018). <https://doi.org/10.1016/j.ifacol.2018.07.064>
- Berbeglia, G., Cordeau, J.F., Laporte, G.: Dynamic pickup and delivery problems. *Eur. J. Oper. Res.* **202**, 8–15 (2010). <https://doi.org/10.1016/j.ejor.2009.04.024>

- Börjesson, M., Eliasson, J.: Experiences from the Swedish Value of Time study. *Transp. Res. A Policy Pract.* **59**, 144–158 (2014). <https://doi.org/10.1016/j.tra.2013.10.022>
- Ceder, A., Wilson, N.H.: Bus network design. *Transp. Res. B Methodol.* **20**, 331–344 (1986)
- Chew, J.S.C., Lee, L.S., Seow, H.V.: Genetic algorithm for biobjective urban transit routing problem. *J. Appl. Math.* **2013**, 1–15 (2013). <https://doi.org/10.1155/2013/698645>
- Cordeau, J.F., Laporte, G.: The dial-a-ride problem: Models and algorithms. *Ann. Oper. Res.* **153**, 29–46 (2007). <https://doi.org/10.1007/s10479-007-0170-8>
- Dantzig, G.B., Ramser, J.H.: The Truck Dispatching Problem. *Manage. Sci.* **6**, 80–91 (1959). <https://doi.org/10.1287/mnsc.6.1.80>
- Derigs, U., Pullmann, M., Vogel, U.: Truck and trailer routing—Problems, heuristics and computational experience. *Comput. Oper. Res.* **40**, 536–546 (2013). <https://doi.org/10.1016/j.cor.2012.08.007>
- Desrochers, M., Lenstra, J., Savelsbergh, M., Soumis, F.: Vehicle routing with time windows: Optimization and approximation. *Veh. Rout. Methods Stud.* **16** (1988)
- DLR: U-Shift II (Demonstrator) (2021). <https://verkehrsforschung.dlr.de/en/projects/u-shift/u-shift-ii-demonstrator>
- Drexl, M.: Applications of the vehicle routing problem with trailers and transshipments. *Eur. J. Oper. Res.* **227**, 275–283 (2013). <https://doi.org/10.1016/j.ejor.2012.12.015>
- Dündar, H., Ömürgöntülgen, M., Soysal, M.: A review on sustainable urban vehicle routing. *J. Clean. Prod.* **285**, 125444 (2021). <https://doi.org/10.1016/j.jclepro.2020.125444>
- Economic Research Institute: Bus Driver Salary Stockholm, Sweden (2024). <https://www.salaryexpert.com/salary/job/bus-driver/sweden/stockholm>
- Ghilas, V., Demir, E., Van Woensel, T.: An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows and scheduled lines. *Comput. Oper. Res.* **72**, 12–30 (2016a). <https://doi.org/10.1016/j.cor.2016.01.018>
- Ghilas, V., Demir, E., Woensel, T.V.: The pickup and delivery problem with time windows and scheduled lines. *INFOR: Inf. Syst. Oper. Res.* **54**, 147–167 (2016b). <https://doi.org/10.1080/03155986.2016.1166793>
- Hatzenbühler, J., Jenelius, E., Gidófalvi, G., Cats, O.: Modular vehicle routing for combined passenger and freight transport. *Transp. Res. A Policy Pract.* **173**, 103688 (2023). <https://www.sciencedirect.com/science/article/pii/S0965856423001088>, <https://doi.org/10.1016/j.tra.2023.103688>
- Huber, S., Geiger, M.J.: Order matters – a variable neighborhood search for the swap-body vehicle routing problem. *Eur. J. Oper. Res.* **263**, 419–445 (2017). <https://doi.org/10.1016/j.ejor.2017.04.046>
- Koç, Ç., Bektaş, T., Jabali, O., Laporte, G.: Thirty years of heterogeneous vehicle routing. *Eur. J. Oper. Res.* **249**, 1–21 (2016). <https://doi.org/10.1016/j.ejor.2015.07.020>
- Lenstra, J.K., Kan, A.H.G.R.: Complexity of vehicle routing and scheduling problems. *Networks* **11**, 221–227 (1981). <https://doi.org/10.1002/net.3230110211>
- Li, B., Krushinsky, D., Reijers, H.A., van Woensel, T.: The Share-a-Ride Problem: People and parcels sharing taxis. *Eur. J. Oper. Res.* **238**, 31–40 (2014). <https://doi.org/10.1016/j.ejor.2014.03.003>
- Li, B., Krushinsky, D., Van Woensel, T., Reijers, H.A.: An adaptive large neighborhood search heuristic for the share-a-ride problem. *Comput. Oper. Res.* **66**, 170–180 (2016). <https://doi.org/10.1016/j.cor.2015.08.008>
- Los, J., Schulte, F., Spaan, M.T.J., Negenborn, R.R.: The value of information sharing for platform-based collaborative vehicle routing. *Transp. Res. E: Logist. Transp. Rev.* **141**, 102011 (2020). <https://doi.org/10.1016/j.tre.2020.102011>
- Marinov, M., Giubilei, F., Gerhardt, M., Özkan, T., Stergiou, E., Papadopol, M., Cabecinha, L.: Urban freight movement by rail. *J. Transp. Lit.* **7**, 87–116 (2013). <https://doi.org/10.1590/S2238-10312013000300005>
- Masson, R., Lehuédé, F., Péton, O.: An adaptive large neighborhood search for the pickup and delivery problem with transfers. *Transp. Sci.* **47**, 344–355 (2013). <https://doi.org/10.1287/trsc.1120.0432>
- Melachrinoudis, E., Ilhan, A.B., Min, H.: A dial-a-ride problem for client transportation in a health-care organization. *Comput. Oper. Res.* **34**, 742–759 (2007). <https://doi.org/10.1016/j.cor.2005.03.024>
- Militão, A.M., Tirachini, A.: Optimal fleet size for a shared demand-responsive transport system with human-driven vs automated vehicles: A total cost minimization approach. *Transp. Res. A Policy Pract.* **151**, 52–80 (2021). <https://doi.org/10.1016/j.tra.2021.07.004>
- Mourad, A., Puchinger, J., Chu, C.: A survey of models and algorithms for optimizing shared mobility. *Transp. Res. B Methodol.* **123**, 323–346 (2019). <https://doi.org/10.1016/j.trb.2019.02.003>
- Mueller-Eberstein, F., Franke, M.: Project CarGo tram, **124**, 337–340 (2000). https://www.researchgate.net/publication/297113963_Project_CarGo_Tram/
- Parragh, S.N.: Introducing heterogeneous users and vehicles into models and algorithms for the dial-a-ride problem. *Transp. Res. Part C Emerg. Tech.* **19**, 912–930 (2011). <https://doi.org/10.1016/j.trc.2010.06.002>

- Parragh, S.N., Cordeau, J.F.: Branch-and-price and adaptive large neighborhood search for the truck and trailer routing problem with time windows. *Comput. Oper. Res.* **83**, 28–44 (2017). <https://doi.org/10.1016/j.cor.2017.01.020>
- Parragh, S.N., Cordeau, J.F., Doerner, K.F., Hartl, R.F.: Models and algorithms for the heterogeneous dial-a-ride problem with driver-related constraints. *OR Spectrum* **34**, 593–633 (2012). <https://doi.org/10.1007/s00291-010-0229-9>
- Pisinger, D., Ropke, S.: A general heuristic for vehicle routing problems. *Comput. Oper. Res.* **34**, 2403–2435 (2007). <https://doi.org/10.1016/j.cor.2005.09.012>
- Pisinger, D., Ropke, S.: Large Neighborhood Search. In: Gendreau, M. (Ed.), *Handbook of Metaheuristics*, 2nd ed. Springer, p. 399420 (2010)
- Qu, Y., Bard, J.F.: The heterogeneous pickup and delivery problem with configurable vehicle capacity. *Transp. Res. Part C Emerg. Tech.* **32**, 1–20 (2013). <https://www.sciencedirect.com/science/article/pii/S0968090X13000612>, <https://doi.org/10.1016/j.trc.2013.03.007>
- Rekiek, B., Delchambre, A., Saleh, H.A.: Handicapped person transportation: An application of the grouping genetic algorithm. *Eng. Appl. Artif. Intell.* **19**, 511–520 (2006). <https://doi.org/10.1016/j.engappai.2005.12.013>
- Ropke, S., Pisinger, D.: An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transp. Sci.* **40**, 455–472 (2006). <https://doi.org/10.1287/trsc.1050.0135>
- Sacramento, D., Pisinger, D., Ropke, S.: An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. *Transp. Res. Part C Emerg. Tech.* **102**, 289–315 (2019). <https://doi.org/10.1016/j.trc.2019.02.018>
- Savelsbergh, M., Woensel, T.V.: 50th Anniversary invited article—city logistics: challenges and opportunities. *Transp. Sci.* (2016). <https://doi.org/10.1287/trsc.2016.0675>
- Savelsbergh, M.W.P., Sol, M.: The general pickup and delivery problem. *Transp. Sci.* **29**, 17–29 (1995). <https://doi.org/10.1287/trsc.29.1.17>
- Scania: NXT concept vehicle represents a vision of the future (2020). <https://www.scania.com/uk/en/home/experience-scania/features/nxt-concept-vehicle-represents-a-vision-of-the-future.html>
- Schröder, S., Liedtke, G.T.: Towards an integrated multi-agent urban transport model of passenger and freight. *Res. Transp. Econ.* **64**, 3–12 (2017). <https://doi.org/10.1016/j.retrec.2016.12.001>
- Shaw, P.: A New Local Search Algorithm Providing High Quality Solutions to Vehicle Routing Problems (1997)
- Tellez, O., Vercraene, S., Lehuédé, F., Péton, O., Monteiro, T.: The fleet size and mix dial-a-ride problem with reconfigurable vehicle capacity. *Transp. Res. Part C Emerg. Tech.* **91**, 99–123. <https://www.sciencedirect.com/science/article/pii/S0968090X18303851>, <https://doi.org/10.1016/j.trc.2018.03.020>
- Todosijević, R., Hanafi, S., Urošević, D., Jarboui, B., Gendron, B.: A general variable neighborhood search for the swap-body vehicle routing problem. *Comput. Oper. Res.* **78**, 468–479 (2017). <https://doi.org/10.1016/j.cor.2016.01.016>
- Toffolo, T.A., Christiaens, J., van Malderen, S., Wauters, T., Vanden Berghe, G.: Stochastic local search with learning automaton for the swap-body vehicle routing problem. *Comput. Oper. Res.* **89**, 68–81 (2018). <https://doi.org/10.1016/j.cor.2017.08.002>
- Toth, P., Vigo, D. (eds.): *Vehicle Routing: Problems, Methods, and Applications*. MOS-SIAM Series on Optimization, 2nd edn. Society for Industrial and Applied Mathematics; Mathematical Optimization Society, Philadelphia (2014)
- VeRoLog: VeRoLog Solver Challenge 2014 – Verolog (2014). <https://www.euro-online.org/websites/verolog/verolog-solver-challenge-2014/>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Jonas Hatzenbühler has a PhD in Transport Planning from the Division of Transport Planning within the Department of Civil and Architectural Engineering at KTH Royal Institute of Technology, Sweden. His research focuses on impact studies of autonomous and modular vehicle concepts. His work aims at creating efficient and sustainable transport systems by utilizing mathematical programming methods.

Erik Jenelius is Professor in Public Transport Systems and Head of the Division of Transport Planning within the Department of Civil and Architectural Engineering at KTH Royal Institute of Technology, Sweden. His research focuses on developing the transportation system towards sustainability and climate neutrality, while also ensuring they are efficient, inclusive, and reliable. His work is dedicated to creating a more resource-efficient system that emphasizes collective travel, micro-mobility, and active modes of transportation.

Győző Gidófalvi is an Associate Professor at the Division of Geoinformatics at the Department for Urban Planning and Environment as well as Leader of the Transport Electrification research program at the Integrated Transport Research Lab (ITRL) at KTH Royal Institute of Technology, Sweden. His main research interests include big mobility data analytics and artificial intelligence for transport infrastructure and operations optimization, travel survey methods, travel behavior modeling, traffic management, spatio-temporal data mining and analysis, Location-Based Services (LBS), locations privacy, Intelligent Transportation Systems (ITS), and related GIS areas such as web and mobile GIS and spatial databases. He is also the co-founder and CEO of Gordian Logistics Optimization Systems.

Oded Cats is a Professor of Passenger Transport Systems, co-director of the Smart Public Transport Lab and Vice Head of the Department of Transport & Planning at Delft University of Technology in the Netherlands. He is also a Guest Professor at the Transport Planning Division at KTH Royal Institute of Technology in Stockholm, Sweden