Transportation Research Part C 156 (2023) 104353

Contents lists available at ScienceDirect

Transportation Research Part C

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

The multi-vehicle dial-a-ride problem with interchange and perceived passenger travel times

K. Gkiotsalitis [a,*], A. Nikolopoulou [b]

[a] University of Twente, Department of Civil Engineering, 7500 AE Enschede, The Netherlands
[b] Athens University of Economics and Business 47A Evelpidon & 33 Lefkados, 11362, Athens, Greece

ARTICLE INFO

Keywords:
Multi-vehicle DARP
VRP with cross-docking
Branch-and-cut
Tabu search

ABSTRACT

The Dial-a-Ride Problem (DARP) introduced in the early 1980s is the NP-Hard optimization problem of developing the most cost-efficient vehicle schedules for a number of available vehicles that have to start from a depot, pick up and deliver a set of passengers, and return back to the same depot. DARP has been used in many modern applications, including the scheduling of demand-responsive transit and car pooling. This study departs from the original definition of DARP and it extends it by considering an interchange point where vehicles can exchange their picked-up passengers with other vehicles in order to shorten their delivery routes and reduce their running times. In addition to that, this study introduces the concept of generalized passenger travel times in the DARP formulation which translates the increased in-vehicle crowdedness to increased perceived passenger travel times. This addresses a key issue because the perceived in-vehicle travel times of passengers might increase when the vehicle becomes more crowded (i.e., passengers might feel that their travel time is higher when they are not able to find a seat or they are too close to each other increasing the risk of virus transmission or accidents). Given these considerations, this study introduces the Dial-a-Ride Problem with interchange and perceived travel times (DARPi) and models it as a nonlinear programming problem. DARPi is then reformulated to a MILP with the use of linearizations and its search space is tightened with the addition of valid inequalities that are employed when solving the problem to global optimality with Branch-and-Cut. For large problem instances, this study introduces a tabu search-based metaheuristic and performs experiments in benchmark instances used in past literature demonstrating the computation times and solution stability of our approach. The effect of the perceived passenger travel times to the vehicle running costs is also explored in extensive numerical experiments.

1. Introduction

The first models for the dial-a-ride problem (DARP) were developed in the early 1980s to plan the routes of buses based on telephone requests from passengers (Psaraftis, 1980). Over the years, the applicability of DARP models has been extended to on-demand services including minibuses, shared modes, and last-mile transport modes (Ho et al., 2018). There are both static and dynamic DARP formulations in literature, however static formulations have received more attention because, in practice, the transportation requests of passengers are known in advance (Borndörfer et al., 1999; Cordeau and Laporte, 2007; Gkiotsalitis, 2023). The fleet availability and size are also known in advance, and vehicles start from and return to a pre-specified depot (Alesiani et al.,

* Corresponding author.
E-mail addresses: k.gkiotsalitis@utwente.nl (K. Gkiotsalitis), anikolop@aueb.gr (A. Nikolopoulou).

https://doi.org/10.1016/j.trc.2023.104353
Received 8 September 2022; Received in revised form 8 September 2023; Accepted 13 September 2023
Available online 22 September 2023

2021). Prospective passengers can impose a time window on both their departure and arrival times from/to their origin/destination locations (Jaw et al., 1986). Typically, DARP formulations impose a maximum limit on the ride times of passengers and the route times of vehicles (Parragh, 2011). The simplest version of DARP considers a single vehicle and can be solved as a dynamic program (Psaraftis, 1980). The main difference of DARP from the pickup and delivery problem (PDP) in logistics is that it considers the ride times of passengers as an additional problem constraint.

The classic DARP formulation aims to develop vehicle schedules where each vehicle starts from the depot, picks up and delivers passengers, and returns to the same depot. In this formulation, passengers cannot transfer between vehicles and they are inside the same vehicle from their embarkment at the pickup location until their disembarkment at the delivery location. Given that service providers seek to reduce the vehicle running times, it is often beneficial to use different vehicles for the pick up and the delivery of passengers. This has led to the recent development of dial-a-ride problems with transfers (DARPT), where passengers are allowed to transfer between vehicles (Deleplanque and Quilliot, 2013). This study considers the case where vehicles start from the depot, pick up passengers, and return them to the depot (called interchange point) where they are re-arranged to (potentially) different vehicles. Then, passengers are delivered to their final destinations by the re-arranged vehicles. The problem description discussed herein has the benefit that a vehicle can be assigned to pick up a group of closely located passengers even if their destination locations are widely dispersed since these passengers will be reassigned to other vehicles at the interchange point. For a graphical illustration of the DARPT the reader is referred to Fig. 3.

Our proposed formulation guarantees that passengers will have at most one transfer at the depot location, called the *interchange point*. Meanwhile, the service provider can reduce its vehicle running time costs by arranging the schedules of the available vehicles more efficiently via splitting them into pickup schedules and delivery schedules. That is, a vehicle can pick up a different set of passengers than the set of passengers that will finally deliver. This problem formulation is beneficial for applications where the depot of the service operator is at a central location which splits the pickup and delivery locations of passengers into two, resulting in a centralized hub-and-spoke network. The main reasoning behind using one transshipment point, lies in the idea of improving vehicle capacity utilization, allowing vehicles to stop for transfers and adjustments of their transport loads, as well as allowing drivers to switch vehicles or get release times for conforming to policy-related matters. We note that hub-and-spoke networks are widely used in fixed-line services given their economy of scale-related benefits compared to point-to-point networks (Fielbaum et al., 2016). In this study, we propose a DARP formulation that applies to DARP problems with a pickup and delivery distribution of passengers which resembles a hub-and-spoke network. We call this problem the dial-a-ride problem with interchange (DARPi) because passengers might change vehicle at the interchange point location.

The DARPi problem description resembles the pickup and delivery problem with a crossdock (PDPCD). Under the PDPCD, a transportation request can be either directly delivered from its origin to its destination by one vehicle or it can be picked up by one vehicle, and then transported to the cross-dock and delivered to its destination by another vehicle. In the proposed network setting, all passengers must pass from the cross-dock; however, only those who will reach their destination point with a different vehicle will need to disembark. Furthermore, there are additional significant differences, such as:

1. the consideration of route duration constraints for pickup and delivery vehicles.
2. the consideration of ride time constraints for passengers.
3. the consideration of the inconvenience of passengers when the in-vehicle occupancy increases beyond a certain level, resulting in overcrowding, in the perceived passenger ride times.

To consider in-vehicle overcrowding, we use perceived travel times that capture the discomfort of passengers when traveling inside a crowded vehicle. This is an important topic because recent studies show that the perception of a passenger's in-vehicle travel time changes with the level of crowdedness (Yap et al., 2020). This is especially true when passengers cannot find a seat, they are too close to each other, or they cannot meet the social distancing requirements during a pandemic (Gkiotsalitis, 2021).

In this work we consider the static DARP case in which the objective function is to minimize the vehicle running costs. Quality of service is enforced by defining appropriate maximum ride times for each user considering the perceived travel times. Our contribution in this paper is threefold:

- first, we formulate the Dial-a-Ride Problem with interchange and generalized travel times that consider in-vehicle crowding (DARPi).
- second, we model this problem as a quadratic integer program and linearize this formulation resulting in a MILP.
- third, we propose a tabu search heuristic in order to find solutions even for large instances.

In addition to the above, we generate test instances for different problem sizes and we test our model on these instances investigating (i) the optimality gap of the tabu search heuristic in small problem instances, (ii) its performance stability in larger problem instances, and (iii) the impact on passenger travel times and vehicle crowdedness when considering perceived travel times in the problem's formulation.

The remainder of our paper is structured as follows. Section 2 provides a literature review on the dial-a-ride and pick up and delivery problems considering their classic forms and their extensions that consider transfers. Section 3 introduces the DARPi formulation when considering the *actual* travel times of passengers. Section 4 expands this formulation by considering the *perceived* travel times of passengers. DARPi is formulated as a mixed-integer non-linear program and it is linearized to a mixed-integer linear (MILP) one, which we show that it an NP-complete decision problem. In Section 5 we introduce a problem specific Tabu search-based solution method equipped with arc-exchange neighborhood structures to compute solutions for larger instances. Section 7 provides the numerical experiments of our study using benchmark instances of up to 200 requests. Finally, future directions are provided in our conclusion section.

2. Literature review

2.1. Dial-a-ride studies

DARP seeks to define a set of minimum cost routes in order to satisfy a set of transportation requests. Each request involves transporting a set of users from a set of origins, called *pickup points*, to a set of destinations, called *delivery points*. Users associated with distinct requests can share the same vehicle as long as the vehicle's capacity is not exceeded. A maximum ride time, corresponding to the maximum duration of the trip between the pickup and the delivery point, is associated with each request. In this section we review relevant literature on the static, multi-vehicle case where time windows, route duration constraints and passenger ride time constraints are simultaneously considered.

One of the first heuristics to solve the DARP was proposed in the work of Jaw et al. (1986). The authors presented a sequential insertion procedure for solving a DARP variant considering only time window and ride time constraints. Several metaheuristic algorithms have been later proposed to solve the DARP. Cordeau and Laporte (2003a) formulated and solved the static DARP with a tabu search heuristic. Their model allows users to specify a time window of a fixed width on their inbound or outbound trips, with an upper limit on the travel time for any user. Their work has later been extended to the dynamic case by Attanasio et al. (2004), where the primary objective is to accept as many requests, received throughout the day, as possible while satisfying operational constraints. The authors developed a number of parallel heuristics based on a Tabu search previously proposed for the static case (Cordeau and Laporte, 2003b). Madsen et al. (1995) were the first to explicitly solve the DARP with multiple objectives. They resort to the weighted sum approach to incorporate cost as well as user-related objectives into their optimization procedure. A classical cluster first, route second algorithm is proposed in Borndörfer et al. (1999) where customer satisfaction is taken into account in terms of punctual service. Customer ride times are implicitly considered by means of time windows. Xiang et al. (2006) present a heuristic for solving a large-scale static dial-a-ride problem bearing complex constraints. Wong and Bell (2006) present a modified insertion heuristic to solve the DARP with multi-dimensional capacity constraints, and the performance of the proposed algorithm is tested via simulation. Recently, Gschwind and Drexl (2019) have considered a DARP variant that aims at finding a set of minimum-cost routes satisfying constraints on vehicle capacity, time windows, maximum route duration, and maximum user ride times. The authors propose an Adaptive Large Neighborhood Search which proves to be very competitive with state of-the-art DARP heuristics regarding solution quality and running time. Comprehensive surveys on the DARP literature are provided by Cordeau et al. (2010), Doerner and Salazar-González (2014), Parragh et al. (2008).

Many papers have also focused on studying a generalization of the DARP in which users can be transferred from one vehicle to another at intermediate points, called transfer or transshipment points. This problem is called the Dial-A-Ride Problem with Transfers (DARPT). Most research works consider multiple transfer points which might correspond to a pickup or drop off location. Deleplanque and Quilliot (2013) consider transfer in multiple transfer points. In this work, the transfer points can be located at any pickup or dropoff node in the network and there is no dedicated transshipment facility for the transfers. In many of these applications, a user is dropped off at a transfer point before he is picked up by another vehicle (Schönberger, 2017). The synchronization of passenger transfers between vehicles is the main challenge in this variant. More specifically, this area of research aims to minimize the impacts of passenger transfers on users' inconvenience (e.g., users' waiting time). Masson et al. (2014) presented a generalization of the DARP where passengers could make transfers at intermediate points and claimed that significant savings could be achieved when transfers are allowed, in terms of total distance traveled.

The literature review reveals a gap in past studies related to manpower requirements for the transportation of people with limited mobility. These requirements could include the preference of assigning the same driver to a user in multiple periods or the users might request to have accompanying staff members on the vehicle. Additional concerns about manpower requirements in DARPs include the loads of the accompanying staff members and the synchronization of vehicles and assistants. In addition, most DARPs consider only new user requests under a dynamic and deterministic environment. These problems do not consider, however, other types of events such as long delays in vehicle arrival times that require modifications of existing plans or affect the synchronization of vehicles. New modeling techniques and frameworks should be proposed to capture these factors related to disruption management. The aforementioned statements are recommended topics for future work. For a more comprehensive survey on research gaps and opportunities in dial-and-ride problems, the interested reader is referred to the work of Ho et al. (2018).

2.2. Fleet scheduling studies involving transfers

Comparable to the DARPT is the Pickup and Delivery Problem with Transfers (PDPT) for the transportation of goods. The first definition is based on the General Pickup and Delivery Problem (PDP) by Savelsbergh and Sol (1995), in which packages are directly shipped from suppliers to their associated customer by the same carrier. Berbeglia et al. (2007) have identified different categories of PDPs where distinction is made in planning based on supplier-customer ratio. Many suppliers delivering to many customers (many-to-many), or a single supplier to a single customer (one-to-one), while keeping in mind the possibilities of shipments between these two examples (one-to-many and others). Pickup and Delivery Problems with intermediate facilities extend classical PDPs introducing the opportunity for a vehicle to drop or pickup loads at an intermediate facility, called cross-dock (or, sometimes, transshipment point) in this context. The introduction of this type of facility may lead to a more efficient use of the vehicle capacities but increases considerably the complexity of the problem. The main issue in such a problem is how to model the operations carried out at the intermediate facility and, in particular, how to model the flows of freight moving from a pickup vehicle to a delivery vehicle. While in PDP the same vehicle fleet is used for performing both pickup and delivery operations, PDPT is an extension that relaxes

the constraint that packages are delivered by the same carrier, enabling the transfer of packages between carriers. The PDPT has predefined transfer locations where carriers may interchange packages. Mitrović-Minić and Laporte (2006) were one of the first that developed a PDPT with a single transfer point. Additional research into solution techniques for PDPT with a single transfer node was performed by Cortés et al. (2010), Masson et al. (2013) and Rais et al. (2014), where the last allows transfers to take place at any node. Cortés et al. (2010) showed PDPT can yield solutions that in broad terms are more efficient when transfers are allowed. The authors presented a mathematical formulation and proposed a solution method based on Benders decomposition to quantify the trade off between the advantage of adding flexibility (by relaxing the coupling constraints) and the extra disutility due to the addition of transfers to the operational scheme. Masson et al. (2014) introduced the transfer of packages with the use of shuttle services between two transfer points. Additionally, Ghilas et al. (2016) modeled transportation of passengers with time windows and synchronization of routes while using scheduled lines for transfers.

Another closely related research area of the literature is the Vehicle Routing Problem with Cross-docking (VRPCD) for the transportation of goods. The work of Wen et al. (2009) has been used as the basis for various papers. This seminal work addressed a practical distribution problem stemming from a Danish company. The authors considered time window constraints on all pickup and delivery nodes of the distribution network as well as on the cross-dock (CD) facility, to model the presence of a fixed planning horizon. In addition, the restriction of the simultaneous arrival of vehicles at the CD was dropped and more elaborate synchronization constraints were considered to properly coordinate the inbound and outbound product flows. The authors presented a Mixed Integer Linear Programming (MILP) formulation and proposed a Tabu Search heuristic embedded within an Adaptive Memory Programming framework to solve problem instances involving up to 200 transportation requests.

Hasani-Goodarzi and Tavakkoli-Moghaddam (2012) studied the VRPCD with split deliveries and multiple products. As in the work of Lee et al. (2006), the authors imposed simultaneous arrival of inbound vehicles at the CD. The problem was formulated as a MILP aiming to determine the best vehicle routes and the optimal number of utilized vehicles. In a later work, Tarantilis (2013) proposed a multi-restart Tabu Search algorithm for solving the benchmark problem instances of Wen et al. (2009) and examined an alternative scenario involving different vehicles for performing the pickup and delivery operations. Furthermore, the author examined total routing costs for open and closed route network configurations. Morais et al. (2014) developed three Iterated Local Search heuristic algorithms for solving the VRPCD. The proposed algorithms were tested on the set of instances introduced by Wen et al. (2009) as well as on a set of randomly generated instances of larger size, involving up to 500 nodes. To solve the VRPCD, Dondo and Cerdá (2013) developed two different solution approaches: (i) a rigorous MILP formulation and (ii) a sweep-heuristic based on a MILP model. The authors considered two alternative objective functions. The first one involved the minimization of the total routing costs, whereas the second one involved the minimization of the return time to the CD of the last outbound vehicle.

Sadri Esfahani and Fakhrzad (2014) proposed two meta-heuristic algorithms based on Tabu search and Variable Neighborhood Search to solve the VRPCD. Birim (2016) also presented a Simulated Annealing algorithm to address the Vehicle Routing Problem in a cross-docking setting with heterogeneous fleet of vehicles. An Adaptive Memory Programming algorithm was developed in the work of Nikolopoulou et al. (2017) for solving large scale problem instances in a cross-docking network with many-to-many relationship between suppliers and customers. Grangier et al. (2017) proposed a metaheuristic based on large neighborhood search for solving the VRPCD. Abad et al. (2018) proposed a bi-objective mathematical programming model to tackle the integrated pickup and delivery pollution-routing problem with consolidation decisions and used a matheuristic based on large neighborhood search. Baniamerian et al. (2019) considered a profitable heterogeneous vehicle routing problem with cross-docking. The authors developed a hybrid meta-heuristic algorithm based on modified Variable Neighborhood Search and a Genetic Algorithm to solve problem instances of large scale. Finally, Zachariadis et al. (2022) introduced a new Vehicle Routing Problem with cross-docking and capacity restrictions at the cross-dock. A mathematical formulation is proposed to capture the business setting of the new problem. Subsequently, a local search-based metaheuristic algorithm was developed for solving medium and large-scale problem instances.

Another research stream in the literature has focused on the study of hybrid cross-docking distribution networks. The work of Petersen and Ropke (2011) was the first one to study a VRPCD considering optional returns at the CD. In a later work, Santos et al. (2013) studied a hybrid network structure with cross-docking, where direct shipping between suppliers and customers was allowed. A hybrid distribution network was also examined in the work of Nikolopoulou et al. (2019). The authors developed a Tabu Search algorithm to compare two alternative distribution strategies, namely direct shipping and cross-docking, in terms of transportation costs incurred when products must be transferred between two sets of origin and destination points. Recently, Gunawan et al. (2021) proposed a matheuristic algorithm for the VRPCD that does not consider time windows for supplier and customer points. For a comprehensive review of distribution networks with consolidation of freight and merging operations, the reader is referred to the work of Guastaroba et al. (2016). It is also worth noting that a number of works have focused on the Pickup and Delivery Problem with Crossdock (PDPCD) which introduces pickup and delivery routes to VRPCD (see Petersen and Ropke, 2011; Santos et al., 2013).

Summarizing, the literature review reveals the following gaps:

- the DARP with an interchange point has not been formulated in past literature
- there is no inclusion of perceived travel times in the DARP formulation
- there is no exact formulation for the DARP with an interchange point that considers perceived travel times

In addition to the above, although the PDP and VRPCD models have been extensively studied in the past, our literature review reveals a gap in the study of hybrid network schemes. A direction for future studies can be a hybrid strategy allowing both direct-shipping vehicle trips as well as product consolidation at the cross-dock as a more cost-effective transportation option. Furthermore, the use of meta-heuristics to solve this category of problems is a popular choice among the research community. The development

Table 1
Classification of the cited articles based on the problem features considered.

Dial-a-Ride problem										
Research Work	DD	MTP	T	D	TW	MO	SD	HF	HU	RT/RD
Psaraftis (1980)[a]				✓		✓				
Jaw et al. (1986)					✓					✓
Madsen et al. (1995)[b]				✓	✓	✓		✓		✓
Cordeau and Laporte (2003b)					✓			✓		✓
Attanasio et al. (2004)				✓	✓			✓		✓
Wong and Bell (2006)					✓	✓		✓	✓	✓
Xiang et al. (2006)[c]					✓	✓		✓	✓	✓
Parragh (2011)					✓			✓	✓	✓
Gschwind and Drexl (2019)					✓					✓
Gkiotsalitis (2021)					✓			✓	✓	✓
Dial-a-Ride with Transfers										
Research Work	DD	MTP	T	D	TW	MO	SD	HF	HU	RT/RD
Deleplanque and Quilliot (2013)[d]		✓	✓		✓					✓
Masson et al. (2014)		✓	✓		✓					✓
Schönberger (2017)[e]			✓					✓		✓
This work			✓		✓			✓		✓
Pickup and Delivery with Transfers										
Research Work	DD	MTP	T	D	TW	MO	SD	HF	HU	RT/RD
Mitrović-Minić and Laporte (2006)	✓	✓	✓		✓					
Cortés et al. (2010)	✓		✓							
Masson et al. (2013)	✓		✓							
Rais et al. (2014)	✓		✓							
Ghilas et al. (2016)	✓		✓							
Vehicle Routing with Cross-docking										
Research Work	DD	MTP	T	D	TW	MO	SD	HF	HU	RT/RD
Lee et al. (2006)			✓					✓	✓	
Wen et al. (2009)			✓		✓				✓	
Liao et al. (2010)[f]			✓							
Hasani-Goodarzi and Tavakkoli-Moghaddam (2012)[f]			✓		✓		✓	✓		
Vahdani et al. (2012)			✓						✓	
Dondo and Cerdá (2013)			✓						✓	
Esfahani and Fakhrzad (2014)[f]			✓		✓					
Tarantilis (2013)			✓		✓			✓	✓	
Moghadam et al. (2014)			✓		✓		✓		✓	
Morais et al. (2014)			✓		✓					
Nikolopoulou et al. (2017)	✓		✓		✓		✓		✓	
Yin and Chuang (2016)			✓					✓	✓	
Yu et al. (2016)			✓					✓	✓	
Baniamerian et al. (2019)			✓		✓				✓	
Grangier et al. (2017)			✓		✓				✓	
Nikolopoulou et al. (2019)			✓				✓		✓	
Birim (2016)			✓					✓	✓	
Abad et al. (2018)[f]			✓		✓					
Baniamerian et al. (2019)			✓		✓			✓	✓	
Zachariadis et al. (2022)[g]			✓		✓	✓		✓	✓	

DD: Direct delivery; MTP: Multiple transfer points; T: Transfer; D: Dynamic; TW: Time windows; MO: Multiple objectives; SD: Split deliveries; HF: Heterogeneous fleet; HU: Heterogeneous users; RD/RT: Ride time and route duration constraints.

[a] Single vehicle.

[b] Multiple types of vehicle capacities.

[c] Driving constraints are considered.

[d] Dynamic transfer point.

[e] A least transfer time is ensured for all passengers at the transfer point.

[f] Multiple products.

[g] Capacity of the cross-dock facility is considered.

of hybrid methods involving cooperative schemes between exact methods and metaheuristics would also be worth exploring. This is because models are becoming more complex and there is a need to capture more and more realistic operational features and the increasing number of user requests.

Table 2
Classification of the cited articles based on the model type and the solution methods proposed.

Research work	Model type	Solver/Solution method
Psaraftis (1980)	–	DP
Jaw et al. (1986)	–	IH
Madsen et al. (1995)	–	IH
Borndörfer et al. (1999)	–	B&C, SP
Cordeau and Laporte (2003b)	–	TS
Attanasio et al. (2004)	–	TS
Lee et al. (2006)	NLP	TS
Mitrović-Minić and Laporte (2006)	–	H
Wong and Bell (2006)	–	IH
Xiang et al. (2006)	–	LS, CG
Cordeau and Laporte (2007)	–	TS
Wen et al. (2009)	MILP	TS
Cortés et al. (2010)	MILP	B&C
Liao et al. (2010)	–	TS
Parragh (2011)	MILP	B&C, NVS
Hasani-Goodarzi and Tavakkoli-Moghaddam (2012)	MILP	GAMS
Vahdani et al. (2012)	NLP	PSO/SA/VNS
Deleplanque and Quilliot (2013)	–	IH
Dondo and Cerdá (2013)	MILP	H
Esfahani and Fakhrzad (2014)	MILP	TS, VNS
Masson et al. (2013)	–	B&C, ALNS
Tarantilis (2013)	–	TS
Masson et al. (2014)	–	ALNS
Morais et al. (2014)	–	ILS
Moghadam et al. (2014)	MINLP	SA, SA/ACO
Rais et al. (2014)	MIP	MIP Solver
Ghilas et al. (2016)	MIP	CPLEX
Nikolopoulou et al. (2017)	–	TS
Yin and Chuang (2016)	NLP	AMABC
Yu et al. (2016)	MILP	SA
Baniamerian et al. (2019)	MILP	GA
Grangier et al. (2017)	MILP	LNS, SPM
Nikolopoulou et al. (2019)	MILP	AMP, TS
Schönberger (2017)		MA
Abad et al. (2018)	MILP	GA/PSO
Birim (2016)	MILP	SA
Baniamerian et al. (2019)	MILP	GA/VNS, SA/ABC
Gschwind and Drexl (2019)	–	ALNS
Gkiotsalitis (2021)	MILP	–
Zachariadis et al. (2022)	MILP	TS
This work	MILP	TS

ABC: Artificial bee colony; ACO: Ant colony optimization; AMP: Adaptive memory programming; AMABC: Adaptive memory artificial bee colony; BBO: Biogeography-based optimization; B&C: Branch and cut; B&P: Branch and price; CG: Column generation; DP: Dynamic programming; GA: Genetic algorithm; H: Heuristic; IH: Insertion heuristic; LS: Local search; MA: Memetic algorithm; MILP: Mixed-integer linear programming; IP: Integer programming; MINLP: Mixed-integer non-linear programming; NLP: Non-linear programming; PSO: Particle swarm optimization; SA: Simulated annealing; SAICA: Self-adaptive imperialist competitive algorithm; SP: Set partitioning; SPM: Set partitioning and matching problem; TS: Tabu search; VNS: Variable neighborhood search.

Tables 1 and 2 provide a classification of the cited articles in the literature with respect to the special features considered and the algorithms used to solve the problems.

3. DARPi with actual passenger travel times

DARPi is formally defined as follows. Let $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ be a directed graph. The vertex set $\mathcal{V}$ is partitioned into $\{\mathcal{O} \cup \mathcal{P} \cup \mathcal{D}\}$ where $\mathcal{O} = \langle o_1, o_2, o_3, o_4 \rangle$ are four copies of the depot, $\mathcal{P} = \langle 1, \dots, n \rangle$ the set of pickup vertices and $\mathcal{D} = \langle n+1, \dots, 2n \rangle$ the set of delivery vertices. In more detail, o_1 symbolizes the start of the vehicle to pickup customers, o_2 the return of the vehicle to the interchange point, o_3 the departure of the vehicle from the interchange point to deliver customers, and o_4 the end of the trip of the vehicle after it has delivered all customers. The locations of o_1, o_2, o_3, o_4 are the same because they all refer to the depot location. If two passenger requests have a common pickup but different delivery location, the corresponding pickup vertex is duplicated. The same applies if two requests have a common delivery location but different pickup locations. On the other side, if the origin-destination pair of several passengers is the same, the request of these passengers can be represented by a pickup and a delivery pair. That is, the pickup and delivery vertices are associated with exactly one origin-destination pair. It follows that we have n requests, where each request is a couple $(i, n+i)$ with $i \in \mathcal{P}$ being the pickup point and $n+i \in \mathcal{D}$ the associated delivery point for this origin-destination pair. The feasible arc set is $\mathcal{A} = \big\{\{(o_1, j) : j \in \mathcal{P}\} \cup \{(i, j) : i \in \mathcal{P}, j \in \mathcal{P}, i \neq j\} \cup \{(i, o_2) : i \in \mathcal{P}\} \cup \{(o_3, j) : j \in \mathcal{D}\} \cup \{(i, j) : i \in$

$\mathcal{D}, j \in \mathcal{D}, i \neq j\} \cup \{(i, o_4) : i \in \mathcal{D}\}\big\}$. Note that a vehicle cannot go directly from a pickup point to a delivery point without passing from the interchange point, denoted by o_2, o_3. This is a key difference to DARP problems with transfers (DARPT) or Pickup-and-Delivery Problems with Transfers (PDPT). In essence, a vehicle starts from o_1, serves pickup nodes belonging to set $\mathcal{P}$, returns to the interchange point o_2 where it exchanges its passengers, and then departs from the interchange point o_3 delivering its newly assigned passengers to their delivery points. Finally, it returns back to the depot o_4. This is shown in Fig. 3 in our demonstration section, where 0 is the location of the depot and corresponds to o_1, o_2, o_3, o_4.

To each vertex $i \in \mathcal{V}$, there is an associated pickup or delivery demand q_i with $q_i \geq 0\ \forall i \in \mathcal{P}$, and $q_i = q_{i-n}\ \forall i \in \mathcal{D}$. This demand represents the number of passengers of the origin-destination pair $(i, n+i)$. Note that vehicles start from o_1 empty and return to o_4 empty. That is, $q_{o_1} = q_{o_4} = 0$. There is also a minimum service duration for boarding/alighting every passenger. If $\beta \in \mathbb{R}_+$ is the fixed time requirement for handling a single passenger, then this duration is $\beta \sum_{i \in \mathcal{P}} q_i$ at the interchange location. For the interchange location, we also assume an additional fixed time for unloading and reloading, $a \in \mathbb{R}_+$.

Let $\mathcal{K}$ be the set of vehicles. The capacity of vehicle $k \in \mathcal{K}$ is denoted as $Q_k \in \mathbb{R}_+$ and the maximum allowed duration of route k as $T_k \in \mathbb{R}_+$. The cost and travel time of traversing a feasible arc $(i, j)\ i \in \mathcal{A}$ without performing intermediate stops is $c_{ij} \in \mathbb{R}_+$ and $t_{ij} \in \mathbb{R}_+$, respectively. Note that the triangular inequality holds because both the costs and the travel times are non-negative. Let $L \in \mathbb{R}_+$ be the maximum allowed ride time of any traveler and $[e_i, l_i]$ the time window within which we should serve vertex i.

Let also $u_i^k \in \mathbb{R}_+$ be the time at which vehicle k starts serving vertex $i \in \mathcal{P} \cup \mathcal{D}$ and $r_i \in \mathbb{R}_+$ the ride time of user i corresponding to request $(i, n+1)$. We also introduce binary variables x_{ij}^k, η_i^k and θ_i^k. x_{ij}^k is equal to 1 if vehicle k serves vertices $(i, j) \in \mathcal{A}$ sequentially, i.e., vertex j is served directly after vertex i. $\eta_i^k = 1$ if vehicle k unloads request $i \in \mathcal{P}$ to the interchange point. $\theta_i^k = 1$ if vehicle k reloads request $i \in \mathcal{P}$ from the interchange point. Binary variables $\tilde{\eta}_k$ and $\tilde{\theta}_k$ indicate also whether vehicle k unloads or reloads at the interchange point, respectively.

The problem has the following continuous, nonnegative variables, the first two of which have been already described:

u_i^k	the time at which vehicle k starts servicing vertex $i \in \mathcal{V}$
r_i	the ride time of traveler i corresponding to request $(i, n+i)$, where $i \in \mathcal{P}$
τ_k	the time at which vehicle $k \in \mathcal{K}$ finishes unloading at the interchange point
w_k	the time at which vehicle $k \in \mathcal{K}$ starts reloading at the interchange point
z_i	the time at which request $i \in \mathcal{P}$ is unloaded at the interchange point

The compact, three-index formulation of the DARPi model is cast below. Note that we use a very large positive number $M \to +\infty$ for modeling purposes. In addition, the nomenclature of the model is provided in the Appendix - Table A.15.

(DARPi with Actual Travel Times):

$$\min \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{A}} c_{ij}^k x_{ij}^k \tag{1}$$

subject to:

$$\sum_{k \in \mathcal{K}} \sum_{j:(i,j) \in \mathcal{A}} x_{ij}^k = 1 \qquad \forall i \in \mathcal{P} \cup \mathcal{D} \tag{2}$$

$$\sum_{i \in \mathcal{P}} \sum_{j:(i,j) \in \mathcal{A}} q_i x_{ij}^k \leq Q_k \qquad \forall k \in \mathcal{K} \tag{3}$$

$$\sum_{i \in \mathcal{D}} \sum_{j:(i,j) \in \mathcal{A}} q_i x_{ij}^k \leq Q_k \qquad \forall k \in \mathcal{K} \tag{4}$$

$$\sum_{j:(o_1,j) \in \mathcal{A}} x_{o_1 j}^k = \sum_{j:(o_3,j) \in \mathcal{A}} x_{o_3 j}^k = 1 \qquad \forall k \in \mathcal{K} \tag{5}$$

$$\sum_{j:(j,o_2) \in \mathcal{A}} x_{j o_2}^k = \sum_{j:(j,o_4) \in \mathcal{A}} x_{j o_4}^k = 1 \qquad \forall k \in \mathcal{K} \tag{6}$$

$$\sum_{i:(i,h) \in \mathcal{A}} x_{ih}^k - \sum_{j:(h,j) \in \mathcal{A}} x_{hj}^k = 0 \qquad \forall h \in \mathcal{P} \cup \mathcal{D}, k \in \mathcal{K} \tag{7}$$

$$u_j^k \geq u_i^k + t_{ij} - M(1 - x_{ij}^k) \qquad \forall (i,j) \in \mathcal{A}, k \in \mathcal{K} \tag{8}$$

$$e_i \leq u_i^k \leq l_i \qquad \forall i \in \mathcal{V}, k \in \mathcal{K} \tag{9}$$

$$\eta_i^k - \theta_i^k = \sum_{j \in \mathcal{P} \cup \{o_2\} : j \neq i} x_{ij}^k - \sum_{j \in \mathcal{D} \cup \{o_4\} : j \neq i+n} x_{i+n,j}^k \qquad \forall i \in \mathcal{P}, k \in \mathcal{K} \tag{10}$$

$$\eta_i^k + \theta_i^k \leq 1 \qquad \forall i \in \mathcal{P}, k \in \mathcal{K} \tag{11}$$

$$\frac{1}{M} \sum_{i \in \mathcal{P}} \eta_i^k \leq \tilde{\eta}_k \leq \sum_{i \in \mathcal{P}} \eta_i^k \qquad \forall k \in \mathcal{K} \tag{12}$$

$$\frac{1}{M} \sum_{i \in \mathcal{P}} \theta_i^k \leq \tilde{\theta}_k \leq \sum_{i \in \mathcal{P}} \theta_i^k \qquad \forall k \in \mathcal{K} \tag{13}$$

$$\tau_k = u_{o_2}^k + a\tilde{\eta}_k + \beta \sum_{i \in \mathcal{P}} q_i \eta_i^k \qquad \forall k \in \mathcal{K} \tag{14}$$

$$w_k \geq \tau_k \qquad \forall k \in \mathcal{K} \tag{15}$$

$$u^k_{o_3} = w_k + a\tilde{\theta}_k + \beta \sum_{i \in \mathcal{P}} q_i \theta^k_i \qquad \forall k \in \mathcal{K} \tag{16}$$

$$w_k \geq z_i - M(1 - \theta^k_i) \qquad \forall i \in \mathcal{P}, k \in \mathcal{K} \tag{17}$$

$$z_i \geq \tau_k - M(1 - \eta^k_i) \qquad \forall i \in \mathcal{P}, k \in \mathcal{K} \tag{18}$$

$$u^k_{o_2} - u^k_{o_1} \leq T_k \qquad \forall k \in \mathcal{K} \tag{19}$$

$$u^k_{o_4} - u^k_{o_3} \leq T_k \qquad \forall k \in \mathcal{K} \tag{20}$$

$$r_i = \sum_{k \in \mathcal{K}} \sum_{j:(j,n+i) \in \mathcal{A}} x^k_{j,n+i} u^k_{n+i} - \sum_{k \in \mathcal{K}} \sum_{j:(j,i) \in \mathcal{A}} x^k_{j,i} u^k_i \qquad \forall i \in \mathcal{P} \tag{21}$$

$$r_i \leq L \qquad \forall i \in \mathcal{P} \tag{22}$$

The objective function (1) strives to minimize the total vehicle running costs. Constraints (2) ensure that each request-related vertex is visited exactly once. Constraints (3) and (4) ensure that the vehicle capacity at the pickup and delivery route is not exceeded. Constraints (5) ensure that the pickup route of each vehicle will start from o_1 and the delivery route of each vehicle will start from o_3. Similarly, constraints (6) ensure that every pickup route of a vehicle will return to o_2 and every delivery route will return to o_4. Constraints (7) ensure that when a vehicle arrives to a pickup or delivery vertex, it should depart from that vertex ensuring the flow conservation. Constraints (8) ensure that if arc $(i, j) \in \mathcal{A}$ is served by vehicle k, then the time at which vehicle k starts servicing vertex j is greater than or equal to u^k_i plus the travel time from i to j. Constraints (9) ensure that a vertex $i \in \mathcal{V}$ will be served within its time window. Let us now explain constraints (10). When

$$\sum_{j \in \mathcal{P} \cup \{o_2\}: j \neq i} x^k_{ij} = 1$$

vehicle k will pickup request i. Similarly, when

$$\sum_{j \in \mathcal{D} \cup \{o_4\}: j \neq i+n} x^k_{i+n,j} = 1$$

vehicle k will deliver request i. Constraints (10) result in the following four cases: (a) if request i is picked up but not delivered by vehicle k, then $\eta^k_i - \theta^k_i = 1$ and because $\eta^k_i + \theta^k_i \leq 1$ (see constraints (11)) we have that $\eta^k_i = 1$ and $\theta^k_i = 0$; (b) if request i is not picked up, but it is delivered by vehicle k, then $\eta^k_i - \theta^k_i = -1$ and because $\eta^k_i + \theta^k_i \leq 1$ we have that $\eta^k_i = 0$ and $\theta^k_i = 1$; (c) if request i is not picked up and not delivered by vehicle k, then $\eta^k_i - \theta^k_i = 0$ and because $\eta^k_i + \theta^k_i \leq 1$ we have that $\eta^k_i = 0$ and $\theta^k_i = 0$; (d) if request i is picked up and delivered by vehicle k, then $\eta^k_i - \theta^k_i = 0$ and because $\eta^k_i + \theta^k_i \leq 1$ we have that $\eta^k_i = 0$ and $\theta^k_i = 0$. Note that cases (c) and (d) have the same result because even if a vehicle picks up a request i it will not unload it to the interchange point ($\eta^k_i = 0$) and will not reload it from the interchange point ($\theta^k_i = 0$) if it will be delivered to $i + n$ by the same vehicle.

Constraints (12) and (13) determine whether vehicle k unloads and/or reloads at the interchange point. Constraints (14) determine the time at which vehicle k finishes unloading at the interchange point, which depends on whether it performs any unloading ($\tilde{\eta}_k$) and on the amount of unloading requests ($\sum_{i \in \mathcal{P}} q_i \eta^k_i$). Constraints (15) ensure that a vehicle starts reloading at an interchange point after it has finished the unloading process. Constraints (16) determine the time at which vehicle k has finished its unloading/reloading operations at the interchange point and it is ready to leave. Constraints (17) ensure that reloading of a request i can start after this request has been unloaded at the interchange point. Constraints (18) ensure that the time a request i is unloaded at the interchange point is later than the time vehicle k, which unloads request i, finished its unloading process. Constraints (17) and (18) combined ensure that if a request is unloaded and reloaded by two different vehicles at the interchange point, the unloading vehicle should have completed its unloading process before the reloading vehicle can reload this request. Constraints (19) and (20) ensure that pickup routes from the start of a trip o_1 to the interchange point o_2 and delivery routes from the interchange point o_3 to the end of the trip o_4 do not exceed the maximum allowed route duration T_k. Constraints (21) and (22) compute the ride time of each request i and ensure that this ride time is less than the maximum allowed ride time L.

In this model, constraints (21) are nonlinear. To linearize them, we introduce variable $\tilde{u}_i$ which indicates the time at which vertex $i \in \mathcal{P} \cup \mathcal{D}$ starts to get serviced. Each vertex $i \in \mathcal{P} \cup \mathcal{D}$ is served by exactly one vehicle and we can exploit this fact to linearize constraints (21). The idea is to replace constraint (21) by $r_i = \tilde{u}_{n+i} - \tilde{u}_i,\ \forall i \in \mathcal{P}$. To do so, we need to force $\tilde{u}_i$ to take the value of $u^{k^*}_i$, where k^* is the vehicle that actually serves vertex i. Let σ^k_{ij} be continuous slack variables. Then, we can write $\tilde{u}_i$ as:

$$\begin{aligned}
&\tilde{u}_i + \sigma^k_{j,i} = u^k_i && \forall (j,i) \in \mathcal{A} : i \in \mathcal{P} \cup \mathcal{D}, \forall k \in \mathcal{K} \\
&\sigma^k_{ji} \leq M(1 - x^k_{j,i}) && \forall (j,i) \in \mathcal{A} : i \in \mathcal{P} \cup \mathcal{D}, k \in \mathcal{K} \\
&\sigma^k_{ji} \geq -M(1 - x^k_{j,i}) && \forall (j,i) \in \mathcal{A} : i \in \mathcal{P} \cup \mathcal{D}, k \in \mathcal{K}
\end{aligned} \tag{23}$$

Constraints (24) together with constraints (23) can now replace the nonlinear constraints (21):

$$r_i = \tilde{u}_{n+i} - \tilde{u}_i \forall i \in \mathcal{P} \tag{24}$$

4. DARPi with perceived passenger travel times

4.1. Mixed-integer linear formulation

In the DARPi formulation with actual travel times of Eqs. (1)–(20),(22)–(24) we considered the actual ride times of passengers. There is solid evidence, however, that passenger travel times are perceived as longer than the actual ones in crowded vehicles (Shen et al., 2016). This is due to the inconvenience caused because of crowding. This inconvenience is associated with multiple aspects such as reduced travel comfort (e.g. inability to travel seated), loss of travel time productivity, safety and security concerns, increased stress, fatigue and anxiety, as well as raised perception of travel time (Tirachini et al., 2013).

Relevant literature on crowding travel time valuations suggests that maximum mean crowding penalty values range from 1.4–1.5 to 2.1–2.5 (Kroes et al., 2014; Haywood and Koning, 2015; Batarce et al., 2016; Li et al., 2017), as indicated by stated-preference experiments. Along the same lines, crowding valuations can be estimated from revealed-preference data, where crowding penalties are derived from passengers' actual travel behavior. Crowding penalty values reported by revealed-preference studies indicate max. values of 1.55–1.95 at the crush capacity limit (Hörcher et al., 2017; Yap et al., 2020).

To model these perceived passenger travel times, we introduce first variables v_i^k that denote the passenger load of vehicle k upon leaving vertex i. This passenger load is $v_{o_1}^k = 0$ at vertex o_1 since all vehicles start their trips empty and $v_{o_3}^k = \sum_{i\in D}\sum_{j:(i,j)\in\mathcal{A}} q_i x_{ij}^k$ at vertex o_3. At any other vertex,

$$v_j^k = v_i^k + q_j \quad \text{if } x_{ij}^k = 1 \quad \forall (i,j)\in\mathcal{A} : i \in \mathcal{P}+\{o_1\}, k\in\mathcal{K} \tag{25}$$

$$v_j^k = v_i^k - q_j \quad \text{if } x_{ij}^k = 1 \quad \forall (i,j)\in\mathcal{A} : i \in \mathcal{D}+\{o_3\}, k\in\mathcal{K} \tag{26}$$

Note that constraints (25) and (26) are conditional constraints. That is, the load upon leaving a pickup vertex j is $v_i^k + q_j$ only if vehicle k uses arc (i,j). That is to say, $x_{ij}^k = 1 \implies v_j^k = v_i^k + q_j,\ \ \forall (i,j)\in\mathcal{A} : i \in \mathcal{P}+\{o_1\}, k\in\mathcal{K}$. It is also worth noting that at the delivery vertices (Eq. (26)), the load upon leaving vertex j is equal to the load at the previous vertex i minus the number of delivered passengers, q_j, at vertex j given that $x_{ij}^k = 1$. This captures the in-vehicle load reduction after serving a delivery vertex. These constraints are linearized as follows.

Lemma 4.1. *Constraints (25)–(26) are equisatisfiable with constraints*

$$v_j^k + s_{ij}^k = v_i^k + q_j \quad \forall (i,j)\in\mathcal{A} : i \in \mathcal{P}+\{o_1\}, k\in\mathcal{K} \tag{27}$$

$$v_j^k + s_{ij}^k = v_i^k - q_j \quad \forall (i,j)\in\mathcal{A} : i \in \mathcal{D}+\{o_3\}, k\in\mathcal{K} \tag{28}$$

$$s_{ij}^k \le M(1-x_{ij}^k) \quad \forall (i,j)\in\mathcal{A}, k\in\mathcal{K} \tag{29}$$

$$s_{ij}^k \ge -M(1-x_{ij}^k) \quad \forall (i,j)\in\mathcal{A}, k\in\mathcal{K} \tag{30}$$

where s_{ij}^k with $i\in\mathcal{V}, j\in\mathcal{V}, k\in\mathcal{K}$ are continuous slack variables.

Proof. To prove the equisatisfiability of the constraints, let us focus on the pickup constraint. If $x_{i,j}^k = 1$ for some (i,j,k) triplet, then constraints (29)–(30) will force s_{ij}^k to be equal to 0, thus $v_j^k = v_i^k + q_j$. Ergo, the condition $x_{ij}^k = 1 \implies v_j^k = v_i^k + q_j,\ \ \forall (i,j)\in\mathcal{A} : i \in \mathcal{P}+\{o_1\}, k\in\mathcal{K}$ is satisfied. If, however, $x_{ij}^k = 0$, then v_j^k can differ from $v_i^k + q_j$ because s_{ij}^k can take any value in $s_{ij}\in[-M,+M]$. Indeed, for $x_{ij}^k = 0$ we have that $s_{ij}\in[-M,+M]$ and v_j^k does not necessarily need to be equal to $v_i^k + q_j$. The same holds for the reformulation of the delivery constraints. ■

Adding the requirement that the passenger load should not exceed the vehicle capacity and should not be negative we have:

$$0 \le v_j^k \le Q_k \quad \forall j\in\mathcal{V}, k\in\mathcal{K} \tag{31}$$

Note that constraints (3) and (4) in the DARPi formulation with Actual Travel Times can be replaced by the boundary conditions $v_{o_1}^k = 0$, $v_{o_3}^k = \sum_{i\in D}\sum_{j:(i,j)\in\mathcal{A}} q_i x_{ij}^k$ and constraints (27)–(31).

Using the in-vehicle passenger load upon leaving a vertex, v_j^k, we can associate the perceived passenger travel times with the in-vehicle crowding level. In more detail, although the perceived passenger travel times can be equal to the actual ones for low crowding levels, there is a trigger point where the increase of the in-vehicle load affects the travel time perception. This point is typically considered to be the point when all seats of the vehicle are occupied and the newly boarded passengers need to stand. During pandemic outbreaks, this point can be the in-vehicle load level beyond which we cannot maintain the recommended social distancing.

Let $0 \le \rho \le 1$ be the parameter value that corresponds to the in-vehicle loading point after which the perception of travel times is affected. For in-vehicle crowding levels in the range of $[0, \rho Q_k]$ for a vehicle k, the perceived passenger travel times are the same as the actual ones. If, however, the in-vehicle load v_i^k of vehicle k upon leaving vertex i is greater than ρQ_k, then the perceived in-vehicle travel time can be augmented by a progressive travel time penalty $\gamma(v_i^k - \rho Q_k)$. We note that $\gamma \ge 0$ and this parameter indicates the marginal perceived travel time increase for one unit of in-vehicle crowding increase beyond the "trigger point" ρQ_k. This results in the piecewise linear function of Fig. 1.

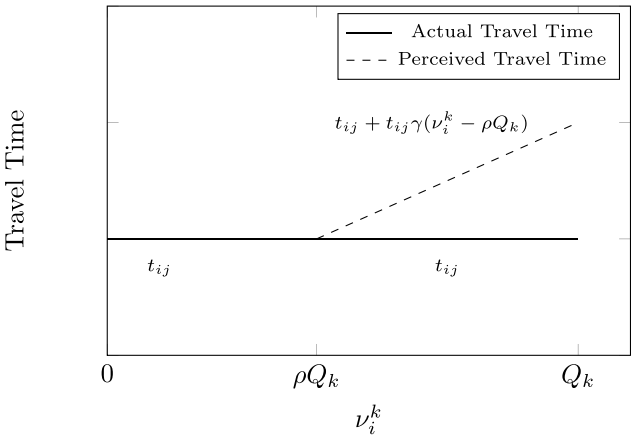

Fig. 1. Perceived vs Actual travel time for different crowding levels, ν_i^k.

From the piecewise linear function, the travel time penalty of vehicle k when leaving vertex i is:

$$d_i^k = 1 + \gamma \max\{\nu_i^k - \rho Q_k, 0\} \quad \forall i \in \mathcal{V}, k \in \mathcal{K} \tag{32}$$

It is important to note that if vehicle i traverses arcs (i,j) after leaving from vertex i, its perceived travel time becomes $t_{ij}d_i^k$. Clearly, if $\nu_i^k \leq \rho Q_k$ then $t_{ij}d_i^k = t_{ij}$ denoting that the perceived travel times of the in-vehicle passengers from i to j will be equal to the actual ones. If, however, $\nu_i^k > \rho Q_k$, there is a perceived travel time $t_{ij}(1 + \gamma(\nu_i^k - \rho Q_k)) = t_{ij} + t_{ij}\gamma(\nu_i^k - \rho Q_k) \geq t_{ij}$.

The travel time penalty d_i^k is provided by a nonlinear function. To linearize it, we introduce binary variables δ_i^k and continuous variables g_i^k. Then, the nonlinear expression (32) is replaced by Eqs. (33)–(38):

$$d_i^k = 1 + \gamma g_i^k \quad \forall i \in \mathcal{V}, k \in \mathcal{K} \tag{33}$$
$$g_i^k \geq \nu_i^k - \rho Q_k \quad \forall i \in \mathcal{V}, k \in \mathcal{K} \tag{34}$$
$$g_i^k \geq 0 \quad \forall i \in \mathcal{V}, k \in \mathcal{K} \tag{35}$$
$$g_i^k \leq \nu_i^k - \rho Q_k + M\delta_i^k \quad \forall i \in \mathcal{V}, k \in \mathcal{K} \tag{36}$$
$$g_i^k \leq M(1 - \delta_i^k) \quad \forall i \in \mathcal{V}, k \in \mathcal{K} \tag{37}$$
$$\delta_i^k \in \{0,1\} \quad \forall i \in \mathcal{V}, k \in \mathcal{K} \tag{38}$$

where constraints (32) and (33)–(38) are equisatisfiable.

To compute the perceived ride times of users, we use the artificial notion of perceived arrival times p_j^k that may differ from the actual arrival times u_j^k at the vertices. Considering the boundary conditions, the perceived time when the service starts at vertex o_1, $p_{o_1}^k$, is equal to $u_{o_1}^k$. When traversing, however, a feasible arc $(i,j) \in \mathcal{A}$ we have a perceived arrival time at j. This conditional equality constraint that provides the value of the perceived arrival time is expressed as:

$$p_j^k = u_j^k + (p_i^k - u_i^k) + t_{ij}d_i^k - t_{ij} \text{ if } x_{ij}^k = 1 \quad \forall (i,j) \in \mathcal{A}, k \in \mathcal{K} \tag{39}$$

The term $t_{ij}d_i^k - t_{ij}$ represents the additional travel time when traversing arc (i,j) due to the in-vehicle crowding upon leaving vertex i. Note that if the in-vehicle load is not beyond ρQ_k, then there is no additional perceived travel time when traversing arc (i,j) because $d_i^k = 1$ and thus $t_{ij}d_i^k - t_{ij} = 0$. Even under this condition, however, the perceived arrival time of vehicle k at vertex j might not be the same as the actual one u_j^k. The reason is that p_i^k might be greater than u_i^k which means that vehicle k had a delayed perceived arrival at vertex i. A special boundary condition is the perceived arrival time from vertex o_3 because there is no travel time from o_2 to o_3 (both are at the location of the depot). This boundary condition is presented in the following equality constraint which is not conditional since all vehicles will unload and reload at the interchange point:

$$p_{o_3}^k = u_{o_3}^k + (p_{o_2}^k - u_{o_2}^k) \quad \forall k \in \mathcal{K} \tag{40}$$

We stress again that these perceived arrival times are an intuitive modeling convention that keeps track of the perceived travel times of passengers and has no physical meaning. Using this convention we can compute exactly the perceived ride time associated with a customer request by replacing constraints (23)–(24) with:

$$\begin{aligned}
&\tilde{p}_i + \sigma_{j,i}^k = p_i^k && \forall (j,i) \in \mathcal{A} : i \in \mathcal{P} \cup \mathcal{D}, \forall k \in \mathcal{K} \\
&\sigma_{ji}^k \leq M(1 - x_{j,i}^k) && \forall (j,i) \in \mathcal{A} : i \in \mathcal{P} \cup \mathcal{D}, k \in \mathcal{K} \\
&\sigma_{ji}^k \geq -M(1 - x_{j,i}^k) && \forall (j,i) \in \mathcal{A} : i \in \mathcal{P} \cup \mathcal{D}, k \in \mathcal{K} \\
&r_i = \tilde{p}_{n+i} - \tilde{p}_i && \forall i \in \mathcal{P}
\end{aligned} \tag{41}$$

Finally, we note that constraints (39) are the only remaining nonlinear constraints due to their conditional form. Let again introduce continuous slack variables $\tilde{s}_{ij}^k$ where $i \in \mathcal{V}, j \in \mathcal{V}, k \in \mathcal{K}$. Then, constraints (39) are linearized as follows:

$$p_j^k + \tilde{s}_{ij}^k = u_j^k + (p_i^k - u_i^k) + t_{ij} d_i^k - t_{ij} \quad \forall (i,j) \in \mathcal{A}, k \in \mathcal{K} \tag{42}$$

$$\tilde{s}_{ij}^k \leq M(1 - x_{ij}^k) \quad \forall (i,j) \in \mathcal{A}, k \in \mathcal{K} \tag{43}$$

$$\tilde{s}_{ij}^k \geq -M(1 - x_{ij}^k) \quad \forall (i,j) \in \mathcal{A}, k \in \mathcal{K} \tag{44}$$

This results in the three-index DARPi formulation that considers the perceived travel times of passengers based on the in-vehicle crowding levels:

(DARPi with Perceived Travel Times):

$$\min \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{A}} c_{ij}^k x_{ij}^k \tag{45}$$

$$\text{subject to: Eqs.(1) – (2), (5) – (20), (22), (27) – (31), (33) – (38), (41) – (44)} \tag{46}$$

$$v_{o_3}^k = \sum_{i \in D} \sum_{j:(i,j) \in \mathcal{A}} q_i x_{ij}^k \tag{47}$$

$$v_{o_1}^k = 0 \tag{48}$$

This formulation is a compact MILP that can be solved to global optimality for small-sized instances with branch-and-cut. The continuous LP relaxations when implementing branch-and-cut can be solved with the polynomial time interior point algorithm of Karmarkar with runtime $O(\omega^{3.5} L)$ where ω is the number of variables. To do so, we need to turn each relaxed LP to its homogeneous form by introducing slack variables such that:

$$\min \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{A}} c_{ij}^k x_{ij}^k$$

$$\text{subject to: } \mathbf{A}\boldsymbol{\omega} = \mathbf{0}$$

$$\mathbf{1}^\intercal \boldsymbol{\omega} = 1$$

$$\boldsymbol{\omega} \geq 0$$

where $\boldsymbol{\omega}$ is a column vector that contains all the variables of the problem, $\mathbf{A}$ an $m \times \omega$ matrix that contains the coefficients of all m constraints of the problem, and $\mathbf{1}$ an ω-valued column vectors of all ones.

Even if the relaxed LPs can be solved in polynomial time, DARPi is NP-Hard because it is a generalization of the pickup and delivery problem with a cross-dock (PDPCD) when considering the passenger ride times and the maximum vehicle route times. Since PDPCD is an NP-complete decision problem (see Santos et al., 2013), DARPi is also NP-complete, and its optimization version is NP-Hard. Thus, one cannot guarantee a globally optimal solution for large problem instances. To obtain solutions for larger instances, in Section 5 we introduce a problem-specific Tabu search equipped with an arc-exchange neighborhood structure.

Finally, we note that the definition of our perceived travel times allows vehicles to perform their operations even if their in-vehicle crowding has to exceed their trigger point ρQ_k due to the high passenger demand and the fixed number of available vehicles. If we were using the trigger point ρQ_k as a hard capacity constraint that cannot be exceeded, the problem would have been infeasible every time at least one of the vehicles had to exceed that limit.

4.2. Valid inequalities

Our DARPi formulation with perceived travel times can become more tight by adding inequality constraints that exclude solutions which cannot be optimal. To do so, these inequality constraints should not eliminate any feasible solutions. That is, if $\mathcal{F}$ is the feasible region of our problem, a valid inequality $\boldsymbol{\pi}\mathbf{x} \leq \pi_0$ should hold for any $\mathbf{x} \in \mathcal{F}$. By making the formulation more tight with the addition of valid inequalities, we increase the computational time of checking the feasibility of potential solutions since additional inequality constraints are added. At the same time, however, the tighter formulation can exclude many solutions that are not optimal resulting in a more focused search process. To tighten the formulation, we add the following valid inequality constraints.

4.2.1. Serve time tightening

$$u_i^k \geq e_i + \sum_{j:(j,i) \in A} \max\{0, e_j - e_i + t_{ij}\} x_{ji} \quad \forall i \in P \cup D, k \in K \tag{49}$$

$$u_i^k \leq l_i + \sum_{j:(i,j) \in A} \max\{0, l_i - l_j + t_{ij}\} x_{ij} \quad \forall i \in P \cup D, k \in K \tag{50}$$

These valid inequalities were used in the past for solving the asymmetric TSP with time windows by branch and cut (Ascheuer et al., 2001).

4.2.2. Arc elimination

With arc elimination we remove arcs that are infeasible. In more detail:

- arc $(i, j) \in A$ is infeasible if $e_i + t_{ij} > l_j$
- arcs (i, j) and $(j', n+i)$ are both infeasible if $t_{i,j} + t_{j,o_2} + t_{o_3,j'} + t_{j',n+i} > L$ for $i \in P, j \in P, j' \in D$

4.2.3. Sub-tour elimination

With sub-tour elimination we add valid inequalities that remove solutions which cannot be optimal, as follows:

- $\sum_{k\in K} x_{ij}^k + \sum_{k\in K} x_{ji}^k \leq 1 \quad \forall i \in P, j \in P$
- $\sum_{k\in K} x_{n+i,n+j}^k + \sum_{k\in K} x_{n+j,n+i}^k \leq 1 \quad \forall i \in P, j \in P$

4.2.4. Ride time: Lower bound

Finally, we add an additional set of valid inequalities related to the lower bounds of ride times to remove solutions that cannot be optimal:

$$r_i^k \geq t_{i,o_2} + (a\tilde{\theta}_k + \beta \sum_{i\in P} q_i \theta_i^k) + t_{o_3,n+i} \quad \forall i \in P, k \in K \tag{51}$$

With these constraints we request that the ride time r_i^k of traveler i corresponding to request $(i, n+i)$ performed by vehicle k should be at least equal to the travel time from i to o_2 plus the travel time from o_3 to $n+i$ plus the time spent during loading and unloading at the interchange point $(a\tilde{\theta}_k + \beta \sum_{i\in P} q_i \theta_i^k)$.

5. Solution method

A local search-based metaheuristic algorithm has been developed, which is capable of deriving high-quality solutions in short computational times for medium and large-scale problem instances. From the algorithmic viewpoint, the proposed framework is initiated by a construction heuristic algorithm that is used to generate an initial solution. Afterwards, the generated solution is improved via a single point trajectory local-search metaheuristic algorithm. In particular, a Tabu Search based scheme is adopted, equipped with arc-exchange neighborhood structures. The construction heuristic algorithm and the Tabu Search scheme are described in Sections 5.1 and 5.2, respectively.

5.1. Construction heuristic algorithm

The proposed construction heuristic algorithm generates routes iteratively by performing minimal cost insertions of service locations. In the beginning, empty routes are initiated, consisting of one pickup and delivery vehicle, each one containing just two occurrences of the depot node, corresponding to the start and end of the route. The number of routes is equal to the number of vehicles performing pickup and delivery operations. At each iteration, the pickup and delivery pair insertion that respects the capacity and time constraints is identified and performed. Only feasible insertion positions are taken into account.

5.2. Tabu search algorithm

A local search metaheuristic algorithm is applied for improving the solution produced by the constructive procedure of Section 5.1. It explores the solution space by performing arc-exchange moves from a current solution to another neighboring solution. These moves are defined by the local search operators presented in Section 5.2.1. The conducted search is diversified by the application of a tabu search based guiding mechanism described in Section 5.2.2. Finally, the overall structure of the local search framework is given in Section 5.2.3.

5.2.1. Local search operators

The neighborhood structures considered are (a) the 0-1 node relocation, (b) the 1-1 node exchange and (c) the 2-Opt. A lexicographic neighborhood evaluation scheme is adopted, while only feasible solutions are considered. The neighborhood structures have quadratic complexity and all of them involve a constant number of edge exchanges. The 0-1 node relocation removes any supplier (or customer) node from its current position and reinserts this node into any other available position of a pickup (or delivery) route. Both intra- and inter-route node relocation moves are considered. The 1-1 node exchange swaps the positions of any two supplier (or customer) nodes served by the same or different pickup (or delivery) routes. Lastly, the 2-Opt exchange move removes a pair of edges and replaces them by a new pair to create the modified solution. More specifically, for any two supplier (or customer) nodes belonging to the same route, the node sequence between them is reversed. On the other hand, if the nodes belong to different routes, then the removal of an edge creates two segments in each route. Two new routes are then created by swapping the route segments following the nodes.

Table 3
Lower and upper time for visiting any vertex $i \in P \cup D$.

i	e_i	l_i	e_{n+i}	l_{n+i}
1	442	562	823	943
2	455	575	852	972
3	360	471	793	913
4	475	595	1007	1127

5.2.2. Diversification component

Our Tabu Search framework operates according to the best admissible local search move scheme. Specifically, all neighborhoods (defined by the operators of Section 5.2.1) of the current incumbent solution are exhaustively explored, and the highest quality feasible neighboring solution is selected. To avoid an over-intensified search, the proposed framework is equipped with a diversification component based on the aspiration criteria of Tabu Search and the Attribute Based Hill Climber (Whittley and Smith, 2004). Each arc $(i, j) \in \mathcal{A}$ is associated with a threshold objective tag t_a. Every time a local-search move mv is applied to solution S with objective value $z(S)$, the threshold tags of the eliminated arcs (E_{mv}) are set equal to $z(s)$, i.e. $t_a = z(s), \forall a \in E_{mv}$. Any move mv that leads to solution S' is considered admissible, only if the cost tags of the generated arcs G_{mv} exceed the objective function value of the modified solution S' (i.e. $t_a > z(S'), \forall a \in G_{mv}$). To control the diversification effect, arc tags are periodically re-initialized to infinity through the search process.

5.2.3. Feasibility evaluation of tentative moves

Whenever a local search move is evaluated, we need to check the vehicle capacity and time window constraints for both pickup and delivery routes. The main assumption is that the pickup routes leave the depot and visit customers as early as possible. All products from the pickup routes are unloaded from the vehicles as soon as these vehicles arrive at the interchange point. The delivery routes are initiated as soon as all necessary products are loaded onto the delivery vehicles.

5.2.4. Overall algorithm structure

The proposed solution method employs a set of checking procedures for evaluating the feasibility of each insertion within the local search. Each of these procedures enables us to check if a given insertion is consistent with a specific category of the problem constraints, by maintaining and frequently updating extra route information. These checks are listed below:

- VehicleCapacity(): Evaluates whether a current move leads to routes that satisfy the vehicle capacity constraints
- BestObjectiveMove(): Evaluates whether a current move improves the best move identified during local search
- Diversification(): Evaluates whether a current move is forbidden according to the diversification component
- RouteDuration(): Evaluates whether a current move leads to routes that respect the route duration constraints
- RideTime(): Evaluates whether a current move leads to routes that respect the ride time constraints
- PRideTime(): Evaluates the perceived ride time of nodes and ensures that the ride time constraints are respected
- TWfeasibility(): Evaluates if a current move leads to delivery routes that respect the customer time window constraints

A pseudocode of the proposed solution framework is presented next (Algorithm 1). Let $z(mv(S))$ refer to the solution objective change brought by the application of move mv at solution S. At each iteration, the steps necessary for identifying the best move for a randomly selected neighborhood structure are presented in Lines 6–13. In particular, the move should lead to feasible vehicle routes w.r.t. vehicle capacity constraints (Line 8) and satisfy the adopted diversification component (Line 9). These moves are then checked for feasibility w.r.t. to the route duration constraints and the ride time constraints (Lines 10, 11). The perceived ride times of suppliers and customers are also evaluated (Line 12). Finally, the TWfeasibility() method is employed to evaluate feasibility of the pickup and delivery routes generated by a move mv considering time windows limitations (Line 13). The selected move is applied, and arc tags are set according to the diversification component (see Section 5.2.3). These tags are randomly reinitialized to infinity according to parameter rst (Lines 16–18). The algorithm is terminated after a certain number of iterations ($iters$) is reached, by returning the lowest objective solution S^{best} encountered.

6. Demonstration

We present the application of our DARPi model in a toy network. The toy network is presented in 2 and it contains $n = 4$ requests. The vehicle capacity is $Q_k = 20$ and we have 2 vehicles. The earliest possible start time of each vehicle is $e_{o_1} = 360$ and the latest possible end time $l_{o_4} = 1320$. Note that all times are in seconds. The fixed time for unloading and reloading at the interchange point is $a = 10$. In addition, the fixed time for handling a single passenger at the interchange point is $\beta = 1$. The maximum allowed duration of a route $T_k = 480\ \forall k \in \{1, 2\}$. The maximum ride time of any traveler is $L = 550$. The passenger demand for pickup/delivery at each vertex is $q_1 = q_5 = 16$, $q_2 = q_6 = 10$, $q_3 = q_7 = 4$, $q_4 = q_8 = 4$.

In addition, Table 3 presents the time window for serving each vertex.

Because the toy network used in this demonstration is small, we can solve it to global optimality with branch-and-cut and Karmarkar's algorithm for the relaxed LPs. Our DARPi formulation is an extension of PDPCD because it considers also the passenger ride times and the maximum vehicle route times. To compare the solution of DARPi against the solution of PDPCD, we first present the results of PDPCD in Section 6.1, and then the results of DARPi with actual and perceived travel times in Sections 6.2 and 6.3, respectively.

Algorithm 1 Pseudocode of the employed tabu search framework that improves initial solution S_0 derived from the construction heuristic

solution $S^{best} \leftarrow S_0, S \leftarrow S_0$
double $thrs, rst$
integer $iters$
$th_a \leftarrow +\infty, \forall a \in E$
while $counter < iters$ **do**
  select at random a local search operator $i \in \{1, 2, 3\}$
  identify the minimum cost local search move mv_i which leads to solution S' such that:
   (a) vehicle capacity constraints are not violated // VehicleCapacity()
   (b) $th_a > z(S'), \forall a \in G_{(mv_i)}$ // Diversification()
   (c) routes are feasible w.r.t. route duration constraints // RouteDuration()
   (d) routes are feasible w.r.t. ride time constraints // RideTime()
   (e) routes are feasible w.r.t. perceived ride time constraints // PRideTime()
   (f) routes are feasible w.r.t. time windows // TWfeasibility()
  $th_a \leftarrow z(S'), \forall a \in E_{mv_i}$
  apply move mv_i
  if $rand(0, 1) < rst$ **then**
    $th_a \leftarrow +\infty, \forall a \in E$
  end if
  if $z(S') < z(S^{best})$ **then**
    $S \leftarrow S'$
    $S^{best} \leftarrow S'$
  end if
end while
return S^{best}

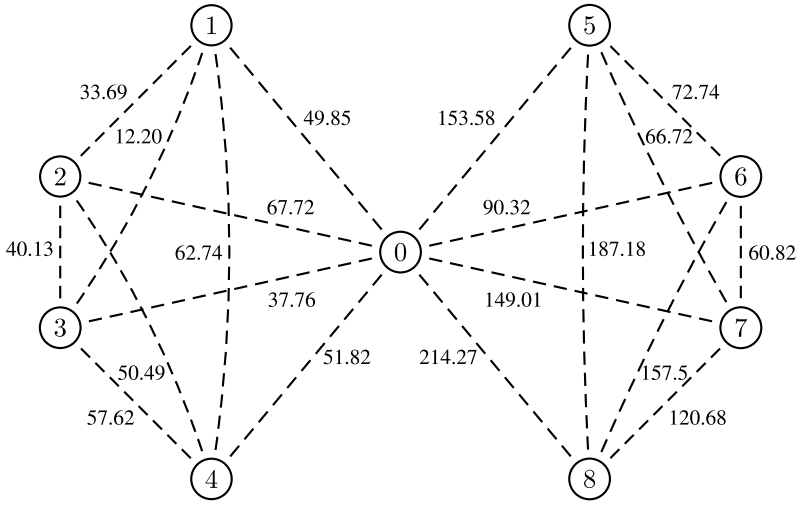

Fig. 2. DARPi Network where vertices o_1, o_2, o_3, o_4 are at location 0 (depot). For visualization purposes, the presented arc travel times t_{ij} are rounded to two decimal places.

6.1. PDPCD results

To compute the solution for the PDPCD which considers parcels instead of passengers, the vehicle travel time and passenger ride time constraints should be inactive. For this we set $L = +\infty$ and $T_k = +\infty \ \forall k \in K$. The optimal pickup and delivery routes of the two vehicles are presented in Fig. 3, where gray color is used to illustrate the routes of the first and black color the routes of the second vehicle. These routes together with their associated travel times are summarized in Table 4.

Finally, the start time of serving each vertex is presented in Table 5. Note that these start times satisfy the imposed time windows. Table 5 reports also the ride time of each request even if we do not impose any constraints to it.

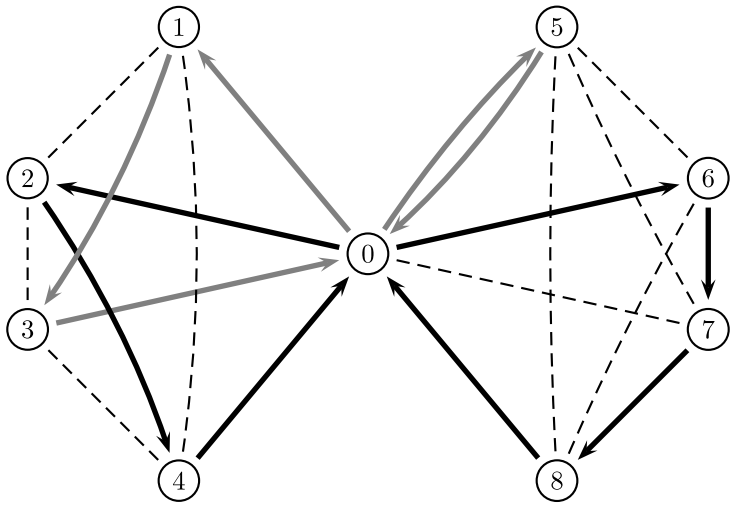

Fig. 3. Optimal pickup and delivery routes of the two vehicles for PDPCD.

Table 4
Travel time of each route and total travel cost for the service provider.

Vehicle	Route type	Served vertices	Route travel time
1	pickup	$o_1 \rightarrow 1 \rightarrow 3 \rightarrow o_2$	99.813
2	pickup	$o_1 \rightarrow 2 \rightarrow 4 \rightarrow o_2$	170.025
1	delivery	$o_3 \rightarrow 5 \rightarrow o_4$	307.154
2	delivery	$o_3 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow o_4$	486.091
total travel cost:			1063.080

Table 5
Start of service time u_i at each vertex $i \in P \cup D$ and passenger ride time r_i for request $i \in P$.

i	e_i	$\tilde{u}_i$	l_i	e_{n+i}	$\tilde{u}_{n+i}$	l_{n+i}	Ride time, r_i
1	442	**442.00**	562	823	**823.00**	943	381.00
2	455	**525.49**	575	852	**852.00**	972	326.51
3	360	**454.20**	471	793	**912.82**	913	458.62
4	475	**475.00**	595	1007	**1105.73**	1127	630.73

Table 6
Travel time of each route and total travel cost for the service provider.

Vehicle	Route type	Served vertices	Route travel time
1	pickup	$o_1 \rightarrow 3 \rightarrow 1 \rightarrow o_2$	99.813
2	pickup	$o_1 \rightarrow 2 \rightarrow 4 \rightarrow o_2$	170.025
1	delivery	$o_3 \rightarrow 7 \rightarrow 5 \rightarrow o_4$	369.310
2	delivery	$o_3 \rightarrow 6 \rightarrow 8 \rightarrow o_4$	462.086
total travel cost:			1101.234

6.2. DARPi results with actual travel times

We now turn our attention to the DARPi problem with actual travel times. To compute its solution, the vehicle travel time and passenger ride time constraints are considered. The optimal pickup and delivery routes of the two vehicles are presented in Fig. 4, where gray color is used to illustrate the routes of the first and black color the routes of the second vehicle. These routes together with their associated travel times are summarized in Table 6.

Finally, the start time of serving each vertex is presented in Table 7. Note that these start times satisfy the imposed time windows. Table 7 reports also the ride time of each request even if we do not impose any constraints to it. There are two main points that we can conclude from this demonstration w.r.t. the behavior of the PDPCD and DARPi:

- the total travel cost of the vehicles with the PDPCD solution is less (1063.080) compared to the total travel cost with the DARPi solution (1101.234). Intuitively, DARPi has additional constraints compared to PDPCD related to the route travel times and the passenger ride times; thus, its solution cannot have a lower objective function score than the solution of the PDPCD.

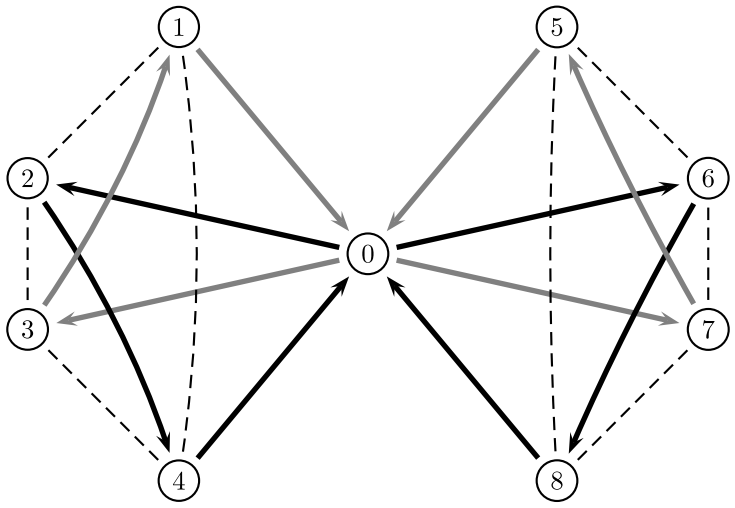

Fig. 4. Optimal pickup and delivery routes of the two vehicles for DARPi with actual travel times.

Table 7
Start of service time u_i at each vertex $i \in P \cup D$ and passenger ride time r_i for request $i \in P$.

i	e_i	u_i	l_i	e_{n+i}	u_{n+i}	l_{n+i}	Ride time, r_i
1	442	**442.0**	562	823	**823.0**	943	381.0
2	455	**544.5**	575	852	**852.0**	972	307.5
3	360	**429.8**	471	793	**913.0**	913	483.2
4	475	**595.0**	595	1007	**1009.5**	1127	414.5

$u^1_{o_1} = u^2_{o_1} = 360$
$u^1_{o_2} = 599.99, u^2_{o_2} = 646.82$
$u^1_{o_3} = 643.99, u^2_{o_3} = 743.77$
$u^1_{o_4} = 1123.99, u^2_{o_4} = 1223.77$.

That is, the total travel vehicle costs of the DARPi solutions are greater than or equal to the total travel vehicle costs of the PDPCD solutions.

- The ride times r_i of the DARPi solution are less than or equal to the maximum allowed ride time $L = 550$, satisfying the ride time constraint. The same holds true for the total route travel times which are less than $T_k = 480$. In contrast, the PDPCD solution had delivery route 2 with travel time $486 > T_k$ and request $i = 4$ with ride time 630.73, which are both greater than L.

6.3. DARPi results with perceived travel times

We solve again the DARPi problem, this time with perceived passenger travel times to explore the potential increase in vehicle running costs when in-vehicle occupancy levels are reduced. We consider $\rho = 0.45$ and $\gamma = 0.1$, resulting in a trigger point of $\rho Q_k = 9$ in-vehicle passengers, after which there is an inconvenience for in-vehicle passengers which is translated to increased perceived travel times. The optimal pickup and delivery routes of the two vehicles are presented in Table 8 and the start times of serving each vertex are presented in Table 9. Notice that the actual ride times have now become perceived ride times and this resulted in a redistribution of pickup passengers. This re-arrangement of passengers at the central interchange point allows vehicles with increased crowding levels to exchange passengers with vehicles having fewer passengers. The perceived ride times are still less than the maximum ride time $L = 550$, however, the total travel cost has increased from 1101.2 to 1111.3 when considering perceived instead of actual running times.

7. Numerical experiments

In this section, we report our computational experience with the Branch-and-cut algorithm and the Tabu search metaheuristic implemented for DARPi with perceived travel times. The two solution methods are tested with instances generated based on the datasets of Wen et al. (2009) introduced for the VRPCD, which are publicly released at https://doi.org/10.11583/DTU.11786763.v1. These datasets are modified and extended because our DARPi formulation requires more data fields related to route duration, ride time constraints, fixed number of vehicles, γ and ρ. The coordinates of the vertices of the instances remain the same as in Wen

Table 8
Travel time of each route and total travel cost for the service provider.

Vehicle	Route type	Served vertices	Route travel time
1	pickup	$o_1 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow o_2$	180.203
2	pickup	$o_1 \rightarrow 1 \rightarrow o_2$	99.702
1	delivery	$o_3 \rightarrow 7 \rightarrow 5 \rightarrow o_4$	369.310
2	delivery	$o_3 \rightarrow 6 \rightarrow 8 \rightarrow o_4$	462.086
total travel cost:			1111.300

Table 9
Start of service time u_i at each vertex $i \in P \cup D$ and perceived passenger ride time r_i for request $i \in P$.

i	e_i	u_i	l_i	e_{n+i}	u_{n+i}	l_{n+i}	Perceived ride time, r_i
1	442	**555.2**	562	823	**859.7**	943	550.0
2	455	**500.8**	575	852	**852.0**	972	468.2
3	360	**441.8**	471	793	**793.0**	913	550.0
4	475	**551.3**	595	1007	**1009.5**	1127	550.0

et al. (2009). All instanced considered in our numerical experiments are made available at https://data.mendeley.com/datasets/trwyhf453c/draft?a=0bebd39a-dc43-4f3a-ad27-efc7ad406b41 and include all Wen et al. (2009) instances ranging from 20 to 200 requests, in addition to smaller instances with 4 to 20 requests used for verification purposes.

Various computational experiments were performed to examine the characteristics of the problem considered as well as to assess the performance of the proposed solution method. Small scale problems were generated to validate the proposed model and solved with both the proposed Tabu search-based solution method and with an branch and cut algorithm. The branch and cut algorithm is implemented with the use of Gurobi 9.0.3 in Python 3.7 and it is publicly released at the aforementioned repository. The Tabu search-based algorithm equipped with arc-exchange neighborhood structures is coded and implemented in C#. Its parameters were initially tuned to improve its performance. We tested values for the rst parameter within the range $[0.002, 0.006]$. The value of 0.003 was selected for the problem instances of smaller scale (from 50 up to 100 requests), while for larger scale problem instances (from 150 up to 200 requests) the value of rst was set equal to 0.004 for which a more effective algorithm behavior was observed. The $thrs$ value was fixed at 0.85 as at this value the algorithm managed to obtain solutions of higher quality within reduced computational times. We used a maximum number of 100,000 iterations as a termination condition for the Tabu search.

For all experiments we used a single thread of a server with an Intel Xeon CPU E5-2650 v2 (2.60 GHz) processor and 16 GB of RAM. Preliminary experiments were conducted to determine a well-performing setting for the input parameters. A time limit of four CPU hours was imposed on the execution of algorithms. This allowed the exact branch and cut approach to solve instances with up to 10 requests.

7.1. Small-scale instances

Small-scale instances were generated because branch-and-cut can be applied in instances with up to 10 requests due to the exponential increase of its computational complexity. Because branch-and-cut solves DARPi instances exactly, we can use the small instances to examine the optimality gap of the Tabu search solutions.

Table 10 provides the results when using branch and cut and tabu search for small instances. Column 2 (CNS) reports the number of constraints of the instance. Column 3 (NE) reports the number of explored nodes until the termination of branch and cut. Column 6 (ost) reports the performance of the globally optimal solution derived by branch and cut. Column 7 (bst) reports the performance of the best solution found by the Tabu search metaheuristic. Because we run Tabu search 10 times for each instance to increase the chance of finding a solution close to the globally optimal one, column 8 (avg) reports the average performance of the best Tabu search solutions in these 10 runs. Column 9 (% diff) reports the difference between the performance of the best and average Tabu search solution. Column 10 (gap %) reports the optimality gap between the performance of the branch and cut and the tabu search solution. Note that this optimality gap is 0% for instances with up to 8 requests. It is also worth noting that Tabu search can solve instances with 10 requests in 11.6 CPU seconds, whereas branch and cut requires more than 3 CPU hours. In five out of seven test problems, the proposed Tabu search-based algorithm managed to find the optimal solution. For the other two problem instances, the gaps of solutions of the Tabu search-based algorithm and the exact model are on average 2.75%. Note that the deviation from the optimal solution for these two instances is because in the metaheuristic algorithm best solutions are provided under the assumption that all the necessary operations are performed as early as possible. More specifically, the pickup routes leave the depot and visit customers as early as possible, the pickup routes are unloaded from the vehicles as soon as these vehicles arrive at the interchange point and the delivery routes are initiated as soon as all necessary products are loaded onto the delivery vehicles (see Section 5.2.3). This assumption is not present in the exact model that is able to explore more options regarding the timing of the pickup/drop-off operations, and thus, provide better solutions.

In order to prove the validity of the proposed algorithm additional computational experiments were performed for the original VRPCD problem as examined by Wen et al. (2009) for small instances. To do this, we have relaxed the route duration constraints and the ride time constraints, as these do not exist in the VRPCD setting. The γ value was set equal to 0. The results confirm that the proposed algorithm performs well and it has managed to match 3 out 5 best solutions found in the literature (see Table 11).

Table 10
Results from small instances with up to 10 requests.

Instance	CNS	Branch and cut			Tabu search				
		NE	CPU (s)	ost	bst	Avg	%diff	CPU (s)	Gap %
4	1091	1	0.1	1101.23	1101.23	1101.23	0.00	1.11	0
5	1536	39	0.4	1115.99	1115.99	1115.99	0.00	1.21	0
6	3082	5,625	3.8	1390.48	1390.48	1390.48	0.00	2.80	0
7	3988	9,892	13.7	1459.75	1459.75	1459.75	0.00	3.47	0
8	5014	70,204	106.1	1556.51	1556.51	1556.51	0.00	6.04	0
9	6160	213,507	1196.3	1518.77	1558.93	1558.93	0.00	6.00	2.64
10	9891	2,611,554	10 917.7	1823.83	1875.83	1875.83	0.00	11.61	2.85

CNS: problem constraints, **NE**: nodes explored by Branch and Cut, **ost**: cost of the globally optimal solution, **bst**: cost of the best solution of the tabu search, **avg**: cost of the average solution of the tabu search, **gap**: optimality gap of the tabu search solution.

Table 11
Comparative results on VRPCD problem instances.

Instance	W		T		M		N		G		LS		TS		
	bst	t	bst	t	bst	t	bst	t	bst	t	bst	t	bst	t	%bks
50a	6471.9	3865	6450.28	28	6453.08	–	6450.28	68	6455.77	149	6450.28	216.42	6450.28	58	0
50b	7410.6	3185	7428.54	15	7434.9	–	7428.54	8	7320.77	240	7428.54	257.22	7428.54	62	1.47
50c	7330.6	3269	7311.77	13	7317.35	–	7311.77	39	7311.77	120	7311.77	286.92	7311.77	68	0
50d	7050.3	3658	7021.39	117	7035.5	–	7028.22	22	7028.69	681	7028.17	239.64	7028.17	73	0.13
50e	7516.8	3159	7451.42	20	7482.01	–	7451.42	32	7452.83	168	7462.33	308.97	7451.42	67	0

Methods: W - Wen et al. (2009); T –Tarantilis (2013); – Morais et al. (2014); N – Nikolopoulou et al. (2017); G – Grangier et al. (2017); LS – Zachariadis et al. (2022); TS - This Paper.
Notation: bst - best solution obtained over multiple runs (W 25 runs; T 3 runs; N 10 runs; G 10 runs; LS 10 runs; TS 10 runs); t - computational time in s (The average time required for generating the final solutions is reported in W and N; All other methods report the total time for obtaining the best solution); %bks - the %deviation between the best TS solution and the best known solution (bks) =100*((LS-bks)/bks).

7.2. Tabu search results on larger instances

The effectiveness of the proposed Tabu search-based solution method in larger instances allows us to use it as a tool for examining the impact of increasing the passenger inconvenience on the total transportation costs, for the basic DARPi model.

Initially, the proposed solution method was applied to the well-studied benchmark instances of Wen et al. (2009) considering different ride time duration levels (*short* and *large*). Table 12 presents the best solution values obtained for each route duration level as well as the % deviation between the best solution scores and the average solution scores obtained over the total number of runs. For the smaller scale problem instances (20 – 30 requests) the increase of the ride time limit of passengers does not have any effect on the total solution cost. As the number of passengers increases, however, we observe a decrease of the total transportation costs when the ride time limit of passengers is increased. The final column of the Table 12 shows that for medium and larger scale problem instances with more than 50 requests, the average decrease in transportation costs is approximately 0.37%.

The rather limited impact of the route duration on the total transportation costs may be attributed to the inherent problem characteristics. The problems of the benchmark dataset exhibit clustered geographic distribution, with the passenger transfer facility being in between the pickup and delivery clusters. Given a solution provided by the proposed algorithm, the cost of the CD adjacent arcs (the distance traveled by vehicles when leaving and returning to the depot) makes up for a large percentage of the total solution cost. Furthermore, the problem instances involve rather narrow time windows for each pickup and delivery point, as well as small vehicle capacities. As a result, the tightness of the time window constraints, the vehicle capacities, as well as the long distances that the vehicles need to travel to depart from or return to the transfer point, do not allow the exploration of cost savings opportunities given the increase of the passengers' ride time.

7.3. Examining the sensitivity of solutions to perceived travel times

To gain more insight on the impact of the increase of the vehicle occupancy on the total transportation costs, we have performed additional computational experiments. The computational results are summarized in Tables 13 and 14.

Table 13 presents the computational results for the shorter ride time level of $L = 850$ and for the cases where (i) the in-vehicle occupancy is not considered (γ is 0) and (ii) the in-vehicle occupancy, and as result the passenger inconvenience, is taken into account (γ is 0.1). It presents the best solution values obtained for each γ value, as well as the percent deviation between scores (column 7) demonstrating the increase in total vehicle running costs when accounting for overcrowding. The results reveal that when increasing the limit of passengers' inconvenience, the transportation costs increase. For medium and larger scale problem instances with more than 50 requests, the average increase in transportation costs is approximately 0.43%.

We also calculate the crowding vehicle penalties for the pickup and delivery operations of vehicles to show the reduction of crowding. In more detail, the crowding vehicle penalty for vehicle k when departing from vertex i is computed as:

$$\tilde{d}_i^k = \max\{v_i^k - \rho Q_k, 0\} \qquad \forall i \in \mathcal{V}, k \in \mathcal{K}$$

Table 12
Results from large instances with up to 200 requests. For each instance we perform 10 runs to report the best and average performances of the Tabu search-based algorithm's solutions.

Instance	L = 850				L = 1100				
	bst	Avg	%diff	CPU(s)	bst	Avg	%diff	CPU(s)	$\%diff^{bst}$
20a	3398.60	3398.60	0.00	6	3398.60	3398.60	0.00	6	0.00
20b	3417.50	3419.92	0.07	4	3416.89	3419.76	0.08	30	−0.02
20c	3303.16	3303.16	0.00	9	3303.16	3303.16	0.00	9	0.00
20d	3646.47	3646.47	0.00	0	3646.47	3646.47	0.00	0	0.00
20e	3209.35	3209.35	0.00	0	3209.35	3209.35	0.00	0	0.00
30a	4509.63	4510.91	0.03	31	4509.63	4510.91	0.03	31	0.00
30b	5013.93	5018.93	0.10	39	5013.93	5018.93	0.10	39	0.00
30c	5123.11	5128.39	0.10	23	5107.78	5108.39	0.01	23	−0.30
30d	4487.22	4488.28	0.02	24	4487.22	4488.28	0.02	24	0.00
30e	5066.04	5070.28	0.08	37	5058.49	5063.05	0.09	26	−0.15
50a	7360.73	7385.07	0.33	95	7331.25	7346.67	0.21	137	−0.40
50b	7865.96	7876.90	0.14	42	7862.33	7870.54	0.10	45	−0.05
50c	7634.50	7635.40	0.01	13	7630.42	7630.42	0.00	10	−0.05
50d	7681.55	7692.46	0.14	28	7672.13	7677.76	0.07	108	−0.12
50e	7796.72	7828.75	0.41	80	7760.43	7768.53	0.10	115	−0.47
100a	14 466.99	14 510.27	0.30	580	14 442.16	14 485.65	0.30	519	−0.17
100b	15 787.54	15 852.62	0.41	204	15 664.93	15 709.88	0.29	272	−0.78
100c	15 074.70	15 093.47	0.12	532	15 042.87	15 085.57	0.28	778	−0.21
100d	15 347.13	15 450.26	0.67	158	15 226.49	15 269.69	0.28	85	−0.79
100e	15 077.87	15 114.58	0.24	642	15 027.57	15 052.97	0.17	191	−0.33
150a	21 208.80	21 269.17	0.28	822	21 110.83	21 187.04	0.36	966	−0.46
150b	22 070.86	22 125.09	0.25	1306	21 988.67	22 028.96	0.18	1551	−0.37
150c	21 262.91	21 321.02	0.27	761	21 169.45	21 239.26	0.33	1183	−0.44
150d	22 089.48	22 210.00	0.55	762	21 966.67	22 039.05	0.33	1113	−0.56
150e	21 088.25	21 166.69	0.37	425	21 106.84	21 120.62	0.07	1782	0.09
200a	28 855.58	28 972.76	0.41	934	28 710.74	28 836.97	0.44	2524	−0.50
200b	29 273.88	29 342.16	0.23	1400	29 194.73	29 288.86	0.32	1068	−0.27
200c	28 224.55	28 275.80	0.18	1021	28 138.12	28 191.00	0.19	2296	−0.31
200d	29 428.14	29 498.82	0.24	1169	29 265.25	29 332.83	0.23	2386	−0.55
200e	28 283.70	28 368.94	0.30	1785	28 163.86	28 224.06	0.21	1358	−0.42
Average			**0.21**				**0.16**		

L: maximum allowed passenger ride time, **bst**: cost of the best solution of the tabu search after 10 runs, **avg**: cost of the average solution of the tabu search from the 10 runs, $\%diff^{bst}$: performance difference of the best solution for $L = 850$ and $L = 1100$.

This penalty is applied to vehicles when moving from one vertex to another and they have more in-vehicle passengers than the crowding limit ρQ_k. In Tables 13 and 14 we present the average in-vehicle crowding penalties for the pick-up (**pVP**) and delivery (**dVP**) vehicles. These key performance indicators together with the total vehicle running costs reported in the best solutions (**bst**) demonstrate the trade-off between (i) reducing the crowdedness of vehicles that is beyond ρQ_k, and (ii) increasing the vehicle running costs.

As expected, the perceived travel times of passengers are greater than the actual travel times, when overcrowding is considered. An interesting observation is that, on average, the actual travel times of passengers as well as the average vehicle penalties due to overcrowding (ATT and PTT), seem to decrease when γ increases. This result might be attributed to the fact that when γ increases, the solutions obtained are expected to be of lower quality in terms of transportation costs. This allows for the exploration of route schemes where the actual duration between a passenger's pickup and drop-off locations is reduced. Furthermore, route based on actual travel times does not penalize crowding levels beyond a certain threshold, allowing more passengers entering a crowded vehicle, if this results in reduced running costs for the overall fleet. As a result, a route based on actual travel times might result in higher perceived travel times and thus causes greater passenger inconvenience.

Table 14 presents the computational results for increased ride times given different in-vehicle occupancy levels (γ is equal to 0, 0,1 or 0.2). The results reveal a marginal increase in transportation costs when considering low levels of in-vehicle occupancy (γ is equal to 0.1) compared to the case where the in-vehicle occupancy is not considered (γ is 0). However, a further increase of the in-vehicle occupancy results to an increase in routing costs, reaching an average of 2.09% for larger scale problem instances. Note that the evaluated increase in routing costs (2.09%) is less sensitive to the type of the network (its lowest value is 1.66% for a network with 200 requests and its highest value is 3.48% for a network with 150 requests). In contrast, for higher values of γ, the average routing costs increase from 0.02% to 2.09% demonstrating that for a given number of vehicles an increase in the perceived overcrowding levels can result in a major rerouting of the vehicles yielding higher routing costs. Hard constraints such as the number of vehicles, the vehicle capacity and the time windows also exist and play a significant role in the total cost of the solution. In addition, the actual travel times of passengers as well as the average vehicle penalties due to overcrowding (ATT and PTT), seem to decrease when γ increases. These results are consistent with those presented in Table 13. Despite the increase in routing costs given a further increase of the in-vehicle occupancy, the actual travel times of passengers decrease. This is an important finding for dial-a-ride companies that seek to increase their customer service level. Regarding the computational times

Table 13
Results in large instances with and without the consideration of perceived passenger travel times for a maximum allowed ride time of $L = 850$.

Instance	$\gamma = 0$ (without perceived travel times)				$\gamma = 0.1$ (with perceived travel times)					
	bst	ATT	pVP	dVP	bst	%diff	ATT	PTT	pVP	dVP
20a	3398.60	465.64	12.75	11.75	3398.60	0.00	465.64	590.94	12.75	11.75
20b	3416.89	381.24	10.50	13.93	3417.50	0.02	374.63	558.85	10.50	13.93
20c	3303.16	406.69	11.10	11.70	3303.16	0.00	406.69	546.19	11.10	11.70
20d	3646.47	374.24	11.25	11.25	3646.47	0.00	374.24	570.50	11.25	11.25
20e	3209.35	413.57	12.67	12.70	3209.35	0.00	413.57	583.83	12.67	12.70
30a	4509.63	407.59	15.83	12.36	4509.63	0.00	407.59	590.90	15.83	12.36
30b	5013.93	396.07	14.40	14.90	5013.93	0.00	396.07	626.34	14.40	14.90
30c	5107.78	388.22	13.30	13.20	5123.11	0.30	377.40	594.19	13.30	13.20
30d	4487.22	408.02	11.75	14.21	4487.22	0.00	408.02	564.53	11.75	14.21
30e	5051.72	407.04	14.60	12.50	5066.04	0.28	413.62	605.50	12.50	12.50
50a	7315.71	407.30	15.08	13.96	7360.73	0.62	406.07	591.09	15.08	13.96
50b	7864.93	396.88	13.31	13.56	7865.96	0.01	407.24	605.32	13.97	13.94
50c	7630.42	379.79	14.90	13.38	7634.50	0.05	376.35	570.49	14.90	13.38
50d	7674.14	415.33	12.90	15.12	7681.55	0.10	413.50	600.24	12.90	13.50
50e	7760.62	398.93	13.88	14.50	7796.72	0.47	397.20	620.66	13.88	13.56
Average	–	**403.10**	**13.21**	**13.27**	–	**0.12**	**402.52**	**587.97**	**13.12**	**13.12**
100a	14 454.27	413.79	14.72	14.43	14 466.99	0.09	408.70	584.78	14.83	13.68
100b	15 652.75	418.28	13.34	14.67	15 787.54	0.86	416.87	607.45	13.47	14.53
100c	15 040.25	407.25	15.26	13.85	15 074.70	0.23	401.77	581.60	14.43	13.09
100d	15 226.49	416.62	13.53	14.72	15 347.13	0.79	416.67	627.31	13.19	14.07
100e	15 039.55	390.84	14.34	13.78	15 077.87	0.25	384.83	571.63	14.21	13.35
150a	21 116.44	391.89	14.30	14.39	21 208.80	0.44	393.12	583.68	14.76	14.05
150b	21 981.51	406.03	14.86	14.57	22 070.86	0.41	404.88	597.82	14.85	14.37
150c	21 190.89	391.66	15.38	14.68	21 262.91	0.34	391.05	592.45	14.77	14.28
150d	21 966.77	415.23	14.50	15.55	22 089.48	0.56	411.91	622.95	14.34	15.17
150e	21 054.56	387.98	14.68	14.68	21 088.25	0.16	387.74	588.46	14.64	14.54
200a	28 719.02	391.52	14.66	14.63	28 855.58	0.48	393.58	596.94	14.66	14.40
200b	29 194.73	397.70	15.35	14.90	29 273.88	0.27	397.34	600.89	14.97	14.95
200c	28 146.05	396.16	14.79	14.97	28 224.55	0.28	393.84	593.23	14.75	14.97
200d	29 244.26	395.21	14.68	14.75	29 428.14	0.63	397.52	605.09	14.58	15.13
200e	28 107.59	395.40	14.48	14.88	28 283.70	0.63	392.73	592.39	14.18	14.66
Average	–	**401.04**	**14.59**	**14.63**	–	**0.43**	**399.50**	**596.44**	**14.44**	**14.35**

γ: marginal perceived travel time increase by passengers for one unit in-vehicle crowding increase beyond the acceptable crowding level ρQ_k, **bst**: cost of the best solution of the tabu search after 10 runs representing the total vehicle running costs, **ATT**: actual travel time of passengers, **PTT**: perceived travel time of passengers, **pVP**: average Pickup Vehicle Penalties due to overcrowding, **dVP**: average Delivery Vehicle Penalties due to overcrowding, %diff: performance difference (%) of the best solution for $\gamma = 0$ and $\gamma = 0.1$.

required to obtain the best solution, we observed a slight increase in CPU times when gamma increases. This was expected and can be attributed to the fact that when γ is greater than 0, additional calculations need to be performed within local search to guarantee the feasibility of tentative moves.

8. Conclusion

In this work we introduced a nonlinear model for the multi-vehicle dial-a-ride problem with interchange and perceived travel times. This model was linearized resulting in a mixed-integer linear programming formulation. We also added valid inequalities to tighten the solution space and we proposed a Tabu search-based metaheuristic equipped with arc-exchange neighborhood structures to solve large problem instances.

Our experiments show that DARPi can be solved to global optimality within a reasonable time for small instances with up to 10 requests. Because of its NP-Hard nature, a Tabu search-based metaheuristic was developed to obtain solutions for larger instances and study their sensitivity to the perceived travel times of passengers. Tabu search was capable of computing solutions for instances with up to 200 requests in less than 30 min. By deploying the DARPi formulation we show that vehicle routing costs might increase by 2.09% if we seek to maintain an appropriate level of in-vehicle crowding that will not lead to excessive perceived passenger travel times. This information is valuable for decision makers and the application of our model for different values of γ can shed light to the trade-off between perceived passenger travel times and the vehicle running costs.

Our study offers an extension to the original DARP formulation by considering an interchange point that can reduce vehicle running costs and perceived passenger travel times. The selection of the interchange point is a critical decision and future research can explore this in more detail with the employment of clustering methodologies. In addition, future research might consider the adoption of multiple interchange points resulting in a more general DARP formulation with multiple transfers. Further, the exact values of perceived travel times in relation to the vehicle crowdedness can be determined in more detail by conducting survey-based experiments that can extract the preferences of the affected passengers. Finally, with the possible introduction of autonomous vehicles in the near future, the proposed routes of our model can be directly transmitted to autonomous vehicles, further automating the process of scheduling with the aim to reduce the perceived travel times of passengers.

Table 14
Results in large instances with and without the consideration of perceived travel times for a maximum allowed ride time of $L = 1100$.

Instance	$\gamma = 0$ (without perceived travel times)				$\gamma = 0.1$ (with perceived travel times)						$\gamma = 0.2$ (with perceived travel times)					
	bst	ATT	pVP	dVP	bst	$\%diff_1$	ATT	PTT	pVP	dVP	bst	$\%diff_2$	ATT	PTT	pVP	dVP
20a	3398.60	465.64	12.75	11.75	3,398.60	0.00	465.64	599.14	12.75	11.75	3,402.17	0.10	468.31	773.49	12.75	11.75
20b	3416.89	369.64	10.50	13.93	3,416.89	0.00	369.64	557.21	10.50	13.93	3,429.79	0.38	379.17	806.61	10.50	13.93
20c	3303.16	414.91	11.10	11.70	3,303.16	0.00	414.91	559.47	11.10	11.70	3,339.58	1.10	409.60	783.49	11.10	11.70
20d	3646.47	371.97	11.25	11.25	3,646.47	0.00	371.97	565.66	11.25	11.25	3,672.47	0.71	374.17	789.04	11.25	11.25
20e	3209.35	411.83	12.67	12.70	3,209.35	0.00	411.83	591.07	12.67	12.70	3,227.48	0.57	412.67	809.17	12.67	12.70
30a	4509.63	412.82	15.83	12.36	4,509.63	0.00	412.82	589.12	15.83	12.36	4,555.36	1.01	405.48	755.22	15.83	10.64
30b	5013.93	393.79	14.40	14.90	5,013.93	0.00	393.79	614.03	14.40	14.90	5,069.83	1.11	394.03	831.86	14.40	12.32
30c	5107.78	383.59	13.30	13.20	5,107.78	0.00	383.59	617.60	13.30	13.20	5,204.35	1.89	378.44	812.66	13.30	11.86
30d	4487.22	408.02	11.75	14.21	4,487.22	0.00	408.02	564.53	11.75	14.21	4,494.25	0.16	418.17	822.88	11.75	14.21
30e	5058.49	408.94	14.60	12.50	5,058.49	0.00	408.94	594.76	14.60	12.50	5,181.41	2.43	402.43	836.63	12.50	12.50
50a	7331.25	402.19	15.08	13.42	7,331.25	0.00	402.19	604.06	15.08	13.42	7,431.40	1.37	407.64	835.68	15.08	13.42
50b	7862.33	403.62	13.31	13.38	7,862.33	0.00	403.62	603.12	13.31	13.38	7,949.04	1.10	402.71	881.79	13.31	13.94
50c	7630.42	380.47	14.90	13.38	7,630.42	0.00	380.47	561.69	14.90	13.38	7,762.74	1.73	382.22	807.28	14.90	13.38
50d	7672.13	413.64	12.90	15.12	7,672.13	0.00	413.64	620.27	12.90	15.12	7,778.63	1.39	407.32	865.65	12.90	13.03
50e	7760.43	400.91	13.88	14.50	7,760.43	0.00	400.91	608.52	13.88	14.50	7,911.01	1.94	397.49	840.60	13.88	12.06
Average	**5293.87**	**402.80**	**13.21**	**13.22**	**5293.87**	**0.00**	**402.80**	**590.02**	**13.21**	**13.22**	**5360.63**	**1.13**	**402.66**	**816.80**	**13.07**	**12.58**
100a	14 442.16	413.69	15.62	14.28	14,442.16	0.00	413.69	590.78	15.62	14.28	14,688.53	1.71	411.22	809.10	14.69	12.74
100b	15 664.93	414.21	14.00	14.31	15,664.93	0.00	414.21	627.75	14.00	14.31	16,011.33	2.21	414.34	848.50	13.34	13.22
100c	15 042.87	407.48	15.05	14.63	15,042.87	0.00	407.48	623.82	15.05	14.63	15,243.44	1.33	407.88	844.01	14.33	13.92
100d	15,226.49	416.62	13.53	14.72	15,226.49	0.00	416.62	652.56	13.53	14.72	15,638.56	2.71	420.43	861.72	13.12	13.02
100e	15 027.57	387.57	13.69	13.78	15,027.57	0.00	387.57	577.28	13.69	13.78	15,306.42	1.86	387.89	831.99	14.34	13.32
150a	21 110.83	392.92	14.34	14.32	21,110.83	0.00	392.92	602.77	14.34	14.32	21,374.03	1.25	390.35	838.10	14.18	13.33
150b	21 988.67	410.01	15.10	14.78	21,988.67	0.00	410.01	615.89	15.10	14.78	22,494.35	2.30	400.09	852.89	14.63	13.67
150c	21 169.45	390.35	15.38	14.99	21,169.45	0.00	390.35	602.28	15.38	14.99	21,735.32	2.67	392.69	835.64	14.26	13.83
150d	21 966.67	413.86	14.36	15.07	21,966.67	0.00	413.86	628.67	14.36	15.07	22,730.07	3.48	405.87	877.84	13.92	12.97
150e	21 059.07	385.96	15.03	15.05	21,106.84	0.23	390.75	616.50	14.68	14.81	21,527.85	1.99	380.66	865.01	14.50	14.07
200a	28 710.74	395.28	14.69	14.58	28,710.74	0.00	395.28	602.61	14.69	14.58	29,328.02	2.15	389.45	847.37	14.31	13.05
200b	29,194.73	397.70	15.35	14.90	29,194.73	0.00	397.70	636.45	15.35	14.90	29,678.33	1.66	396.73	872.92	14.68	14.02
200c	28 138.12	391.30	15.19	15.34	28,138.12	0.00	391.30	597.99	15.19	15.34	28,637.39	1.77	396.18	842.83	14.19	13.89
200d	29 249.20	396.64	14.58	15.21	29,265.25	0.05	396.69	611.96	14.83	14.90	29,932.41	2.28	394.27	865.88	14.21	13.56
200e	28 161.98	395.22	14.78	14.83	28,163.86	0.01	395.29	611.64	14.78	14.50	28,728.44	2.00	394.18	838.98	14.18	13.82
Average	**21743.57**	**400.59**	**14.71**	**14.72**	**21747.95**	**0.02**	**400.92**	**613.26**	**14.71**	**14.66**	**22203.63**	**2.09**	**398.82**	**848.85**	**14.19**	**13.49**

γ: marginal perceived travel time increase by passengers for one unit in-vehicle crowding increase beyond the acceptable crowding level ρQ_k, **bst**: cost of the best solution of the tabu search after 10 runs representing the total vehicle running costs, **ATT**: actual travel time of passengers, **PTT**: perceived travel time of passengers, **pVP**: average Pickup Vehicle Penalties due to overcrowding, **dVP**: average Delivery Vehicle Penalties due to overcrowding, $\%diff_1$: performance difference (%) of the best solution for $\gamma = 0$ and $\gamma = 0.1$, $\%diff_2$: performance difference (%) of the best solution for $\gamma = 0.1$ and $\gamma = 0.2$.

CRediT authorship contribution statement

K. Gkiotsalitis: Conceptualization, Methodology, Data curation, Software, Writing – original draft, Writing – review & editing. **A. Nikolopoulou:** Conceptualization, Methodology, Data curation, Software, Writing – original draft, Writing – review & editing.

Data availability

We have stored our data and software code to Mendeley and we have cited it in our paper.

Appendix A. Table of notations

See Table A.15.

Table A.15
Nomenclature.

Sets	
$\mathcal{P} = \langle 1, \dots, n \rangle$	the set of pickup vertices
$\mathcal{D} = \langle n+1, \dots, 2n \rangle$	the set of delivery vertices
$\mathcal{O} = \langle o_1, o_2, o_3, o_4 \rangle$	four copies of the depot
$\mathcal{V}$	set of all vertices
$\mathcal{A}$	the feasible arc set
$\mathcal{K}$	set of vehicles
Parameters	
q_i	pickup or delivery demand of vertex i
β	a fixed time requirement for handling a single passenger
a	fixed time for unloading and reloading
Q_k	capacity of vehicle k

(*continued on next page*)

Table A.15 (*continued*).

T_k	maximum allowed duration of route k
c_{ij}	cost of traversing a feasible arc (i,j) $i \in \mathcal{A}$ without performing intermediate stops
t_{ij}	travel time of traversing a feasible arc (i,j) $i \in \mathcal{A}$ without performing intermediate stops
L	maximum allowed ride time of any traveler
$[e_i, l_i]$	time window within which we should serve vertex i
Variables	
x_{ij}^k	binary variable equal to 1 if vehicle k serves vertices $(i,j) \in \mathcal{A}$ sequentially, i.e., vertex j is served directly after vertex i
η_i^k	binary variable equal to 1 if vehicle k unloads request $i \in \mathcal{P}$ to the interchange point
θ_i^k	binary variable equal to 1 if vehicle k reloads request $i \in \mathcal{P}$ from the interchange point
$\tilde{\eta}_k$	binary variable indicating whether vehicle k unloads at the interchange point
$\tilde{\theta}_k$	binary variable indicating whether vehicle k reloads at the interchange point.
u_i^k	the time at which vehicle k starts servicing vertex $i \in \mathcal{V}$
r_i	the ride time of traveler i corresponding to request $(i, n+i)$, where $i \in \mathcal{P}$
τ_k	time at which vehicle $k \in \mathcal{K}$ finishes unloading at the interchange point
w_k	time at which vehicle $k \in \mathcal{K}$ starts reloading at the interchange point
z_i	time at which request $i \in \mathcal{P}$ is unloaded at the interchange point

References

Abad, H.K.E., Vahdani, B., Sharifi, M., Etebari, F., 2018. A bi-objective model for pickup and delivery pollution-routing problem with integration and consolidation shipments in cross-docking system. J. Clean. Prod. 193, 784–801.

Alesiani, F., Ermis, G., Gkiotsalitis, K., 2021. Constrained clustering for the capacitated vehicle routing problem (CC-CVRP). Appl. Artif. Intell..

Ascheuer, N., Fischetti, M., Grötschel, M., 2001. Solving the asymmetric travelling salesman problem with time windows by branch-and-cut. Math. Programm. 90 (3), 475–506.

Attanasio, A., Cordeau, J.-F., Ghiani, G., Laporte, G., 2004. Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. Parallel Comput. 30 (3), 377–387.

Baniamerian, A., Bashiri, M., Tavakkoli-Moghaddam, R., 2019. Modified variable neighborhood search and genetic algorithm for profitable heterogeneous vehicle routing problem with cross-docking. Appl. Soft Comput. 75, 441–460.

Batarce, M., Muñoz, J.C., de Dios Ortúzar, J., 2016. Valuing crowding in public transport: Implications for cost-benefit analysis. Transp. Res. Part A: Policy Prac. 91, 358–378.

Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., Laporte, G., 2007. Static pickup and delivery problems: A classification scheme and survey. Top 15 (1), 1–31.

Birim, Ş., 2016. Vehicle routing problem with cross docking: A simulated annealing approach. Procedia-Soc. Behav. Sci. 235, 149–158.

Borndörfer, R., Grötschel, M., Klostermeier, F., Küttner, C., 1999. Telebus berlin: Vehicle scheduling in a dial-a-ride system. In: Computer-Aided Transit Scheduling. Springer, pp. 391–422.

Cordeau, J.-F., Iori, M., Laporte, G., Salazar González, J.J., 2010. A branch-and-cut algorithm for the pickup and delivery traveling salesman problem with LIFO loading. Networks 55 (1), 46–59.

Cordeau, J.-F., Laporte, G., 2003a. The dial-a-ride problem (DARP): Variants, modeling issues and algorithms. Q. J. Belgian French Italian Oper. Res. Soc. 1 (2), 89–101.

Cordeau, J.-F., Laporte, G., 2003b. A Tabu search heuristic for the static multi-vehicle dial-a-ride problem. Transp. Res. B 37 (6), 579–594.

Cordeau, J.-F., Laporte, G., 2007. The dial-a-ride problem: Models and algorithms. Ann. Oper. Res. 153 (1), 29–46.

Cortés, C.E., Matamala, M., Contardo, C., 2010. The pickup and delivery problem with transfers: Formulation and a branch-and-cut solution method. European J. Oper. Res. 200 (3), 711–724.

Deleplanque, S., Quilliot, A., 2013. Dial-a-ride problem with time windows, transshipments, and dynamic transfer points. IFAC Proc. Vol. 46 (9), 1256–1261.

Doerner, K.F., Salazar-González, J.-J., 2014. Chapter 7: Pickup-and-delivery problems for people transportation. In: Vehicle Routing: Problems, Methods, and Applications, Second Ed. SIAM, pp. 193–212.

Dondo, R., Cerdá, J., 2013. A sweep-heuristic based formulation for the vehicle routing problem with cross-docking. Comput. Chem. Eng. 48, 293–311.

Esfahani, A.S., Fakhrzad, M., 2014. Modeling the time windows vehicle routing problem in cross-docking strategy using two meta-heuristic algorithms. International Journal of Engineering 27 (7), 1113–1126.

Fielbaum, A., Jara-Diaz, S., Gschwender, A., 2016. Optimal public transport networks for a general urban structure. Transp. Res. B 94, 298–313.

Ghilas, V., Demir, E., Van Woensel, T., 2016. The pickup and delivery problem with time windows and scheduled lines. INFOR: Inform. Syst. Oper. Res. 54 (2), 147–167.

Gkiotsalitis, K., 2021. The dial-a-ride problem considering the in-vehicle crowding inconvenience due to COVID-19. In: 2021 IEEE International Intelligent Transportation Systems Conference. ITSC, IEEE, pp. 3746–3751.

Gkiotsalitis, K., 2023. Public Transport Optimization. Springer Nature.

Grangier, P., Gendreau, M., Lehuédé, F., Rousseau, L.-M., 2017. A matheuristic based on large neighborhood search for the vehicle routing problem with cross-docking. Comput. Oper. Res. 84, 116–126.

Gschwind, T., Drexl, M., 2019. Adaptive large neighborhood search with a constant-time feasibility test for the dial-a-ride problem. Transp. Sci. 53 (2), 480–491.

Guastaroba, G., Speranza, M.G., Vigo, D., 2016. Intermediate facilities in freight transportation planning: A survey. Transp. Sci. 50 (3), 763–789.

Gunawan, A., Widjaja, A.T., Vansteenwegen, P., Vincent, F.Y., 2021. A matheuristic algorithm for the vehicle routing problem with cross-docking. Appl. Soft Comput. 103, 107163.

Hasani-Goodarzi, A., Tavakkoli-Moghaddam, R., 2012. Capacitated vehicle routing problem for multi-product cross-docking with split deliveries and pickups. Procedia-Soc. Behav. Sci. 62, 1360–1365.

Haywood, L., Koning, M., 2015. The distribution of crowding costs in public transport: New evidence from Paris. Transp. Res. Part A: Policy Prac. 77, 182–201.

Ho, S.C., Szeto, W.Y., Kuo, Y.-H., Leung, J.M., Petering, M., Tou, T.W., 2018. A survey of dial-a-ride problems: Literature review and recent developments. Transp. Res. B 111, 395–421.

Hörcher, D., Graham, D.J., Anderson, R.J., 2017. Crowding cost estimation with large scale smart card and vehicle location data. Transp. Res. B 95, 105–125.

Jaw, J.-J., Odoni, A.R., Psaraftis, H.N., Wilson, N.H., 1986. A Heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. Transp. Res. B 20 (3), 243–257.

Kroes, E., Kouwenhoven, M., Debrincat, L., Pauget, N., 2014. Value of crowding on public transport in île-de-France, France. Transp. Res. Rec. 2417 (1), 37–45.
Lee, Y.H., Jung, J.W., Lee, K.M., 2006. Vehicle routing scheduling for cross-docking in the supply chain. Comput. Ind. Eng. 51 (2), 247–256.
Li, H., Gao, K., Tu, H., 2017. Variations in mode-specific valuations of travel time reliability and in-vehicle crowding: Implications for demand estimation. Transp. Res. Part A: Policy Prac. 103, 250–263.
Liao, C.-J., Lin, Y., Shih, S.C., 2010. Vehicle routing with cross-docking in the supply chain. Expert systems with applications 37 (10), 6868–6873.
Madsen, O.B., Ravn, H.F., Rygaard, J.M., 1995. A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. Ann. Oper. Res. 60 (1), 193–208.
Masson, R., Lehuédé, F., Péton, O., 2013. An adaptive large neighborhood search for the pickup and delivery problem with transfers. Transp. Sci. 47 (3), 344–355.
Masson, R., Lehuédé, F., Péton, O., 2014. The dial-a-ride problem with transfers. Comput. Oper. Res. 41, 12–23.
Mitrović-Minić, S., Laporte, G., 2006. The pickup and delivery problem with time windows and transshipment. INFOR: Inform. Syst. Oper. Res. 44 (3), 217–227.
Moghadam, S.S., Ghomi, S.F., Karimi, B., 2014. Vehicle routing scheduling problem with cross docking and split deliveries. Computers & chemical engineering 69, 98–107.
Morais, V.W., Mateus, G.R., Noronha, T.F., 2014. Iterated local search heuristics for the vehicle routing problem with cross-docking. Expert Syst. Appl. 41 (16), 7495–7506.
Nikolopoulou, A.I., Repoussis, P.P., Tarantilis, C.D., Zachariadis, E.E., 2017. Moving products between location pairs: Cross-docking versus direct-shipping. European J. Oper. Res. 256 (3), 803–819.
Nikolopoulou, A.I., Repoussis, P.P., Tarantilis, C.D., Zachariadis, E.E., 2019. Adaptive memory programming for the many-to-many vehicle routing problem with cross-docking. Oper. Res. 19 (1), 1–38.
Parragh, S.N., 2011. Introducing heterogeneous users and vehicles into models and algorithms for the dial-a-ride problem. Transp. Res. C 19 (5), 912–930.
Parragh, S.N., Doerner, K.F., Hartl, R.F., 2008. A survey on pickup and delivery problems. J. für Betriebswirtschaft 58 (2), 81–117.
Petersen, H.L., Ropke, S., 2011. The pickup and delivery problem with cross-docking opportunity. In: International Conference on Computational Logistics. Springer, pp. 101–113.
Psaraftis, H.N., 1980. A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. Transp. Sci. 14 (2), 130–154.
Rais, A., Alvelos, F., Carvalho, M.S., 2014. New mixed integer-programming model for the pickup-and-delivery problem with transshipment. European J. Oper. Res. 235 (3), 530–539.
Sadri Esfahani, A., Fakhrzad, M., 2014. Modeling the time windows vehicle routing problem in cross-docking strategy using two meta-Heuristic algorithms. Int. J. Eng. 27 (7), 1113–1126.
Santos, F.A., Mateus, G.R., Da Cunha, A.S., 2013. The pickup and delivery problem with cross-docking. Comput. Oper. Res. 40 (4), 1085–1093.
Savelsbergh, M.W., Sol, M., 1995. The general pickup and delivery problem. Transp. Sci. 29 (1), 17–29.
Schönberger, J., 2017. Scheduling constraints in dial-a-ride problems with transfers: A metaheuristic approach incorporating a cross-route scheduling procedure with postponement opportunities. Public Transp. 9 (1), 243–272.
Shen, X., Feng, S., Li, Z., Hu, B., 2016. Analysis of bus passenger comfort perception based on passenger load factor and in-vehicle time. SpringerPlus 5 (1), 1–10.
Tarantilis, C.D., 2013. Adaptive multi-restart tabu search algorithm for the vehicle routing problem with cross-docking. Optim. Lett. 7 (7), 1583–1596.
Tirachini, A., Hensher, D.A., Rose, J.M., 2013. Crowding in public transport systems: Effects on users, operation and implications for the estimation of demand. Transp. Res. Part A: Policy Prac. 53, 36–52.
Vahdani, B., Tavakkoli-Moghaddam, R., Zandieh, M., Razmi, J., 2012. Vehicle routing scheduling using an enhanced hybrid optimization approach. Journal of Intelligent Manufacturing 23, 759–774.
Wen, M., Larsen, J., Clausen, J., Cordeau, J.-F., Laporte, G., 2009. Vehicle routing with cross-docking. J. Oper. Res. Soc. 60 (12), 1708–1718.
Whittley, I.M., Smith, G.D., 2004. The attribute based hill climber. J. Math. Model. Algorithms 3 (2), 167–178.
Wong, K.I., Bell, M.G., 2006. Solution of the dial-a-ride problem with multi-dimensional capacity constraints. Int. Trans. Oper. Res. 13 (3), 195–208.
Xiang, Z., Chu, C., Chen, H., 2006. A fast heuristic for solving a large-scale static dial-a-ride problem under complex constraints. European J. Oper. Res. 174 (2), 1117–1139.
Yap, M., Cats, O., Van Arem, B., 2020. Crowding valuation in urban tram and bus transportation based on smart card data. Transp. A: Transp. Sci. 16 (1), 23–42.
Yin, P.-Y., Chuang, Y.-L., 2016. Adaptive memory artificial bee colony algorithm for green vehicle routing with cross-docking. Applied Mathematical Modelling 40 (21-22), 9302–9315.
Yu, V.F., Jewpanya, P., Perwira Redi, A., 2016. Open vehicle routing problem with cross-docking. Computers & Industrial Engineering 94, 6–17.
Zachariadis, E.E., Nikolopoulou, A.I., Manousakis, E.G., Repoussis, P.P., Tarantilis, C.D., 2022. The vehicle routing problem with capacitated cross-docking. Expert Syst. Appl. 196, 116620.