TU Delft

Delft University of Technology

Automated taxis' dial-a-ride problem with ride-sharing considering congestion-based dynamic travel times

Liang, Xiao; Correia, Gonçalo Homem de Almeida; An, Kun; van Arem, Bart

DOI
10.1016/j.trc.2020.01.024

Publication date
2020

Document Version
Final published version

Published in
Transportation Research Part C: Emerging Technologies

Citation (APA)
Liang, X., Correia, G. H. D. A., An, K., & van Arem, B. (2020). Automated taxis' dial-a-ride problem with ride-sharing considering congestion-based dynamic travel times. *Transportation Research Part C: Emerging Technologies*, *112*, 260-281. https://doi.org/10.1016/j.trc.2020.01.024

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

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

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

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' – Taverne project

https://www.openaccess.nl/en/you-share-we-take-care

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.

Transportation Research Part C 112 (2020) 260–281

Contents lists available at ScienceDirect

Transportation Research Part C

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

Automated taxis' dial-a-ride problem with ride-sharing considering congestion-based dynamic travel times

Xiao Liang[a,*], Gonçalo Homem de Almeida Correia[a], Kun An[b,c], Bart van Arem[a]

[a] Department of Transport & Planning, Delft University of Technology, Delft, the Netherlands
[b] College of Transportation Engineering, Tongji University, Shanghai, China
[c] Institute of Transport Studies, Department of Civil Engineering, Monash University, Melbourne, Australia

ARTICLE INFO

Keywords:
Automated vehicles
Dial-a-ride problem
Dynamic travel time
Rolling horizon
Ride-sharing
Lagrangian relaxation

ABSTRACT

In this paper, we study the dial-a-ride problem of ride-sharing automated taxis (ATs) in an urban road network, considering the traffic congestion caused by the ATs. This shared automated mobility system is expected to provide a seamless door-to-door service for urban travellers, much like what the existing transportation network companies (TNC) do, but with decreased labour cost and more flexible relocation operations due to the vehicles' automation. We propose an integer non-linear programming (INLP) model that optimizes the routing of the ATs to maximize the system profit, depending on dynamic travel times, which are a non-linear function of the ATs' flows. It is important to involve traffic congestion in such a routing problem since for a growing number of ATs circulating in the city their number will lead to delays. The model is embedded within a rolling horizon framework, which divides a typical day into several horizons to deal with the real-time travel demand. In each horizon, the routing model is solved with the demand at that interval and assuring the continuity of the trips between horizons. Nevertheless, each horizon model is hard to solve given its number of constraints and decision variables. Therefore, we propose a solution approach based on a customized Lagrangian relaxation algorithm, which allows identifying a near-optimal solution for this difficult problem. Numerical experiments for the city of Delft, The Netherlands, are used to demonstrate the solution quality of the proposed algorithm as well as obtaining insights about the AT system performance. Results show that the solution algorithm can solve the proposed model for hard instances. Ride-sharing makes the AT system more capable to provide better service regarding delay time and the number of requests that can be attended by the system. The delay penalty on the profit objective function is an effective control parameter on guaranteeing the service quality while maintaining system profitability.

1. Introduction

An automated vehicle (AV), also known as a driverless car and a self-driving car is an advanced type of vehicle that can drive itself on existing roads. SAE International (2014) identifies six levels of driving automation from level 0 (no automation) to level 5 (full automation). Vehicles in full automated mode are not only able to monitor the driving environment and execute the dynamic driving tasks (e.g. steering, braking, responding to events, determining when to change lanes), but also capable to do so in all driving environments (e.g. expressway merging, high-speed cruising, low-speed traffic congestion). Since driving automation is expected to

* Corresponding author.
E-mail address: x.liang@tudelft.nl (X. Liang).

https://doi.org/10.1016/j.trc.2020.01.024
Received 20 March 2019; Received in revised form 24 January 2020; Accepted 25 January 2020
0968-090X/ © 2020 Elsevier Ltd. All rights reserved.

bring significant benefits such as higher safety, lower traffic congestion, lower transport costs, etc. (Hoogendoorn et al., 2014; KPMG, 2012), AVs are predicted to be increasingly used in the future (Nieuwenhuijsen et al., 2018).

Regarding the use of AVs as transit systems, one of their potential applications is to provide automated taxi (AT) service, therefore offering a seamless door-to-door service within the urban area for all passengers (Hyland and Mahmassani, 2018; Liang et al., 2016; Wen et al., 2018). Generally, conventional taxi services are expensive. With the advent of automation, using AVs in a taxi system may create a cheaper type of urban mobility by avoiding extra human costs in driving vehicles (Krueger et al., 2016). Another potential application is to use AVs as part of carsharing systems. Traditional carsharing systems provide more sustainable urban mobility compared to private cars (Shaheen et al., 1999). Vehicles in these systems have higher utilization rates when compared to the privately-owned ones (Celsor and Millard-Ball, 2007; Jorge et al., 2015; Li et al., 2016; Ma et al., 2017; Schuster et al., 2005). However, the shared-use vehicles must be relocated between different areas due to the imbalance in demand, which leads to time and monetary costs (Angelopoulos et al., 2018). Moreover, traditional carsharing systems usually have either fixed vehicle stations for location-based systems or random parking locations for free-floating systems (Balac et al., 2019; Huang et al., 2018). Hence, the users must walk to reach the vehicles. Using AVs in a carsharing system could reduce the vehicles' relocation costs and eliminate the users' self-serve access to the vehicles. Therefore, shared ATs are expected to be as flexible and convenient as traditional taxis and as sustainable and economical as carsharing. In the future, AVs may replace private human-driven vehicles accounting for the majority of people's daily trips (Nieuwenhuijsen et al., 2018). Despite the possible advantages of this new transport system, the traffic congestion that it may cause cannot be ignored, as predicted in previous research.

Private or public ride-sharing is another important component in shared mobility, which aims to bring together travellers who have similar itineraries and time schedules to share rides (Agatz et al., 2012, 2011; Correia and Viegas, 2011; Schaller, 2018). The large demand and the low occupancies in private transport in peak hours create traffic congestion in many urban areas. Ride-sharing allows people to use transport capacity more efficiently (Furuhata et al., 2013). In the conventional ride-sharing system, users can provide a ride as a driver or ask for a ride as a passenger. Once the travel requests are submitted, there will be a matching between the drivers and the riders. In the matching process, the key constraint is the time schedules of the rides. The drivers should have sufficient time flexibility since they need to accomplish the pick-up and drop-off of the passengers and then arrive at their own destinations. If ATs are used in the service scheme of ride-sharing, they will provide the opportunity to transform the role of the drivers into passengers, who have no need to stay in the vehicles for the whole ride. Currently, ride-sharing is happening for example with Uber-pool systems whereby a person may request a ride at a lower price but be willing to share with other passengers.

In this paper, an optimization model and a solution algorithm are proposed to address the problem of assigning ATs to clients and define their routes on an urban road network. The model considers traffic congestion by incorporating travel times that vary with the flow of the ATs. The flow-dependent travel time is handled by a classic Bureau of Public Roads (BPR) function, while the design of the vehicles' routes is related to a dial-a-ride problem (DARP) in Operations Research. The model also allows ride-sharing, in order to increase the transport efficiency of the AT system. Moreover, we foresee a system that can serve real-time requests which become known during the routing process thus requiring new decisions to be done along the day. Therefore, a rolling horizon framework is proposed in which the DARP with dynamic travel times is solved over several horizons while adapting the supply to the real-time requests that keep popping up throughout the day. A customized Lagrangian algorithm is developed to consecutively solve the proposed NP-hard routing problem at each horizon for real case-study applications.

The main contributions of this paper are: firstly, we formulate an optimization model to solve the AT's DARP considering the flow-dependent travel times in the network, allowing for ride-sharing between passengers who have aligned origins and destinations; secondly, we develop a customized Lagrangian relaxation algorithm within a rolling horizon framework which is able to approach the near-optimal solution to the proposed model; thirdly the application of a case study reveals the potential effects of ATs on traffic congestion and provides insights about the AT system performance.

The paper is organized as follows. Firstly, we review the literature done regarding ATs and the DARP in Section 2. Then we introduce the mathematical model for ATs' DARP with dynamic travel times and ride-sharing in Section 3. Next, the rolling horizon framework is described to deal with the real-time requests in Section 4. Section 5 presents the Lagrangian relaxation-based solution algorithm. Then the application to the case-study city of Delft is presented in Section 6. The paper ends with conclusions in Section 7.

2. Literature review

In this section, we first present the literature related to the state of art research on the topic of using vehicle automation in public transport and derive the need for the present study. Then we discuss the DARP which inspires the model that we propose in this paper. The traffic assignment problem is also reviewed in order to incorporate the traffic congestion in the DARP problem.

2.1. Automated taxis

Some researchers have investigated the effects of using AVs on urban transport. Two methods are widely used to test these impacts: (1) agent-based simulation; (2) mathematical optimization. Martinez and Viegas (2017) used agent-based simulation to build a model to test the introduction of 100% automated fleets of ride-sharing taxis to satisfy transport demand in a city. Results showed that with the subway still in operation, each AV could remove 9 out of 10 cars in the case-study city of Lisbon if a maximum 5 min waiting time is allowed; whilst without the subway, the number reduces to 5 out of 10 removed by one AV. Fagnant and Kockelman (2014) used a similar method to study the implications of shared ATs and compared them to conventional vehicle ownership and use. Their results indicate that each shared AV could replace around 11 conventional vehicles, but they add up to 10%

more travel distance. Moreover, Fagnant and Kockelman (2018) used agent-based simulations to optimize the fleet size of AVs in dynamic ride-sharing. The results suggest that dynamic ride-sharing reduces average service times and travel costs for AV users. Spieser et al. (2014) used an analytical mathematical formulation to estimate the number of shared AVs to replace all modes of personal transportation in the case-study city of Singapore. They conclude that a shared-vehicle mobility solution could meet the personal mobility demand of the entire population with 1/3 of the number of passenger vehicles currently being used. Based on all their results above, ATs show potential benefits in urban transport and could replace numerous conventional vehicles while providing the same transport capacity.

Other research has been focusing on using AVs to provide transport for the first/last mile of high capacity public transport trips. Liang et al. (2016) proposed two integer linear programming (ILP) models to study the design of the service area of ATs to satisfy people's first/last mile demand for train trips. By applying the models to Delft-Zuid station in The Netherlands, they concluded that the fleet size, type of motor (electric or conventional), and the size of the service area influence the profitability of such an AT system. On the behaviour modelling side, Yap et al. (2016) positioned AVs as egress mode of train trips and explored the travellers' preferences for ATs. The authors applied a stated preference experiment to estimate a discrete choice model and concluded that travellers' attitudes regarding vehicle automation are far from optimistic. However, as referred, the system they studied did not include AVs being used as conventional taxis in a city, meaning for all origin and destination pairs, which limits their findings for the purpose of informing this research.

It seems that previous research is mainly focused on a relatively small fleet of AVs, or while considering a large fleet for all kinds of trips excluding the impacts of AVs on travel times. A previous study addressed the problem of modelling the AVs' influence on traffic congestion in their routing, however, in this case looking at privately-owned AVs. Correia and van Arem (2016a) proposed a model to route private family-owned AVs in a user equilibrium perspective with the objective of minimizing each family's transport costs. Since it is a non-linear problem, the model was tackled by an iterative process. In a later work, Levin (2017) also considered the traffic congestion effects when studying the routing of a large number of shared AVs by using a link-transmission model. The model was applied to a small network due to the considerable computation time of the method. These two papers propose methods to route private or shared AVs when considering the influence of traffic congestion caused by the vehicles themselves. However, they both need the demand to be known before the solving process, which does not fit a system with real-time requests. Liang et al. (2018) proposed an optimization model to satisfy both reserved and real-time requests in an urban road network with dynamic travel times. They linearized the BPR function thus making the model solvable. Nevertheless, the precision of the modelled travel times is quite low because it is based on a few discrete breakpoints in the BPR function and even with that simplification the computation time is still quite high when using commercial solvers.

In general, there are plenty of studies about using AVs as taxis or as a shared mode of transport. However, little attention has been devoted to using an optimization method to analyse the impact of traffic congestion when routing a real-sized fleet of AVs as taxis in real-scale road networks. A great number of vehicles will inevitably lead to traffic congestion in some parts of the road network, which is relevant for transport planning and transport engineering. Moreover, travelling information from the vehicles can be used for smart routing. For that smart routing to be possible at a city scale, efficient solving algorithms to find good solutions are required. In some cases, heuristics are used to find a feasible solution with the disadvantage of not knowing how far that solution is from the global optimum. In this paper, we aim for finding a good solution for the problem so that we can get insights about the system functioning and at the same time we want to be able to say how good that solution is. Therefore, a mathematical model with an efficient solution algorithm is proposed, which is able to handle the large-scale application of ATs, considering congestion, producing route choices and associated link volumes resulting from the AT trips.

2.2. Dial-a-ride problem

The model we propose in this paper is related to the vehicle routing problem (VRP), which is to design the best routes to provide services from a depot to some customers distributed in the network (Laporte, 2009). In fact, the VRP can be seen as a class of problems since it has many variations based on the diversity of operating conditions and constraints when being applied in practice. Beyond the classical formulation, the most relevant variation to the VRP for this paper is the DARP, which involves transporting people from their origins to their destinations (Ho et al., 2018). What we intend to formulate is a capacitated DARP with request time windows. A capacitated DARP can have vehicular constraints for the seating capacity and when there is more than one seat, it is possible for the passengers to share a ride with others.

In real-world applications, an important dimension of VRP is the availability of information (Psaraftis, 1988). If the assumed inputs to the VRP do not change during the solving period, or during the implementation period of routing results, this routing problem is defined as a static VRP. On the contrary, a dynamic VRP deals with a problem in which the "inputs may (and, generally, will) change (or be updated) during the implementation of the algorithm and the eventual implementation of the route" (Psaraftis, 1980). This type of VRP is also referred to as an online or real-time problem. Static DARP assumes that all passengers' requests are pre-known to the implementation of the algorithm that solves it. In a dynamic DARP case, a new customer request is eligible for consideration at the time it appears, even if it is later than the start of the vehicles' operation (Cordeau and Laporte, 2007). This matches what is required from the AT service being studied in this paper. In a dynamic DARP problem, vehicle routes are redefined from time to time, which requires technological support for the real-time information exchange between the vehicles and the operation centre, e.g. the position and the occupancy of the vehicles (Pillac et al., 2013). A human-driven taxi may be more difficult to route since the decisions are made involving the vehicle status information, the taxi driver and the operation centre. In an AT case, the process should be simplified due to the absence of the human drivers and their corresponding stochastic decisions.

In general, one of the methods to transfer a static VRP to a dynamic one is periodic re-optimization (Pillac et al., 2013). Periodic re-optimization is to return to the solving procedure each time the demand is updated. This update can be an event occurrence (a new request appears, or another one leaves), which is defined as event trigger; while it can also be a pre-determined duration, of which the most commonly used one is called the rolling horizon. The rolling horizon framework solves the dynamic VRP by decomposing the problem's time dimension and generating a series of static VRP. Since the planning horizon is divided into multiple small periods of time, it is possible, in some cases, to use exact methods. This is also a way to handle the NP-hardness of the problem while abdicating from finding a global optimum for the whole period of optimization, which would in many cases not be possible anyway because the inputs are revealed along the day. Yang et al. (1999) presented a rolling horizon framework for the truck fleet assignment and scheduling problem when dealing with real-time information. Luo and Schonfeld (2011) compared the rolling horizon strategy with immediately inserting requests in the dial-a-ride problem. They concluded, from their computation results, that when satisfying all the demand, the rolling horizon strategy reduced up to 10% the number of vehicles when compared with the immediate strategy. This is because the rolling horizon strategy benefits from having information available in advance.

The problem that we are studying differs significantly from previous research on dynamic DARP due to the large-scale network application. Generally, traditional VRP, including DARP, is focusing on tracking routes, which reflects the visiting sequences of each vehicle. In addition, it assumes that the travel times between a fixed pair of nodes are deterministic regardless of the number of routing vehicles, even though those travel times may change along the day according to some pre-determined patterns. This assumption remains in most practical applications of ATs since the fleet is small compared with the number of other vehicles driving in the same network. In this paper, we are considering an AT system with a larger number of vehicles, meaning that the dynamic traffic effect should not be ignored. To optimize the AT's routing under congestion, we introduce the formulation of traffic assignment to the road network integrated with the DARP.

Traffic assignment is a method to distribute car trips on a road network taking into account the congestion effects of the route choices of the drivers. A mathematical programming framework proposed by Beckmann et al. (1955) is commonly used as a congested assignment technique. The formulation is a non-linear programming problem due to the functions that represent the traffic flow and capture the complex interactions among vehicles. This can be solved in two main ways: linearizing the problem through simplifications or using iterative algorithms. An iterative algorithm for solving the problem starts with an initial set of link costs and link flows and updates those by successive all-or-nothing assignments. Correia and van Arem (2016) used this principle to tackle the incremental process of assigning each private automated vehicle to the road network. The method avoids the non-linearity of the cost-flow function inside mathematical programming. Instead, the travel times resulting from a BPR function are updated after the assignment process of all the vehicles in each iteration. In this paper, we design an iterative process to converge the travel times, which are a function of the traffic flow, as a way to solve the non-linearity of Beckmann's formulation.

3. Model description

In this section, we present an integer non-linear programming (INLP) model to design the optimal routes of a shared AT system. This first version is to be solved statically requiring the demand to be known in advance.

3.1. Problem setting

The notations used in this section are presented in Table 1.

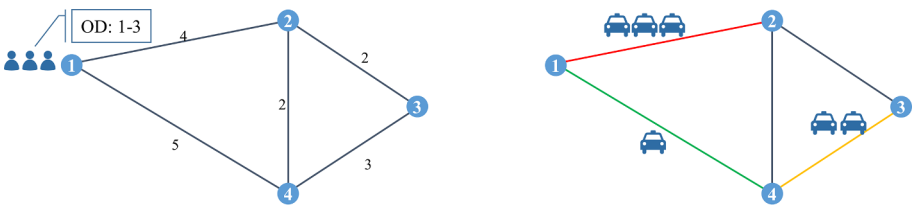
The model works on the assumption that the taxi company can achieve total control of the system by being free to accept or reject requests according to a profit maximization objective. The AT transport service can serve any pair of nodes within the city road network. The travel demand is generated between the nodes with desired time windows. Pick-up and drop-off activities only happen in each time instant meaning that there will be no such activities considered during the time steps (between two sequential time instants). In Fig. 1 we show a small example to illustrate how this system works. The values on each link show the shortest travel time (free-flow travel time) for that link. In this case, two passengers plan to travel from node 1 to node 3 (Fig. 1 (a)). For these two passengers, the shortest path would be 1-2-3 with travel time 6 in total. However, the shortest travel time is not applicable all the time since the travel time will be influenced by how many ATs are travelling on that link. When traffic congestion happens on link 1-2 of the shortest path 1-2-3 (Fig. 1 (b)), the travel time will increase, which makes other possible paths like 1-4-3 competitive or even shorter than path 1-2-3. Therefore, an optimization model is needed to decide which paths the ATs should choose to satisfy clients' travel requests based on the dynamic travel time, which is defined as a function of traffic flow.

We consider a future scenario in which the ATs will replace all modes of personal transport thus the alternative travel modes will be mass public transport e.g. metro, bus and tram. Since these alternative transport modes are usually seen as not contributing to the congestion in the network, we do not consider background traffic flow for simplification meaning that the flow is generated only by the ATs themselves. The model individualizes the vehicles instead of treating them as flows. Parking is not allowed which means that all the ATs should be cruising all the time pro-actively relocating to demand areas. The reason for making this assumption is to apply the iterative assignment solving process, which will be further explained in Section 5.1. The AT system allows several clients to be pooled together respecting vehicle capacity and travellers' schedules. If the travel request cannot be fulfilled, then a penalty will occur: we assume that the AT company would have to pay compensation which can be viewed as paying partially the cost of an alternative transport like metro or bus.

Demand is pre-known for the whole optimization period in this section, meaning that passengers submit their travel requests before the optimization period. When they reserve an AT by an online app, they submit their travel information, i.e. the origin, the

Table 1
Notations.

Notation	Description
Sets	
$\boldsymbol{I}$	$= \{1, \cdots, i, \cdots I\}$, set of nodes in the network, where I is the total number of nodes.
$\boldsymbol{T}$	$= \{0, \cdots, t, \cdots, T\}$set of time instants in the optimization period, where T is the total number of time steps in the operation time. We use time instants to describe the instantaneous state of the AT system, where between two sequential time instants is one time step.
$\boldsymbol{E}$	$= \{1, \cdots, e, \cdots, E\}$, set of travel requests, where E is the total number of passengers' requests in the optimization period.
$\boldsymbol{V}$	$= \{1, \cdots, v, \cdots, V\}$, set of vehicles, where V is the total number of taxis in the system.
$\boldsymbol{G}$	$= \{\cdots, (i, j), \cdots\}$, set of links in the network where i and j are adjacent nodes, $i, j \in \boldsymbol{I}$, $i \neq j$.
$\boldsymbol{M}$	$= \{\cdots, (i_{t_1}, j_{t_2}), \cdots\}$, set of links in the time-space network, $\forall (i, j) \in \boldsymbol{G}$, $\forall t_1, t_2 \in \boldsymbol{T}$, $t_1 < t_2$, $\delta_{ij}^{min} \leq t_2 - t_1 \leq \delta_{ij}^{max}$.
Parameters	
a^e	desired departure time for the eth travel request, $\forall e \in \boldsymbol{E}$.
w^e	maximum waiting time for the eth travel request, $\forall e \in \boldsymbol{E}$.
m^e	origin node for the eth travel request, $\forall e \in \boldsymbol{E}$.
n^e	destination node for the eth travel request, $\forall e \in \boldsymbol{E}$.
opt^e	shortest travel time in time steps for the eth travel request, $\forall e \in \boldsymbol{E}$.
lon^e	longest travel time in time steps for the eth travel request, $\forall e \in \boldsymbol{E}$.
δ_{ij}^{max}	maximum travel time in time steps on the link from node i to node j, $\forall (i, j) \in \boldsymbol{G}$.
δ_{ij}^{min}	minimum travel time in time steps on the link from node i to node j, $\forall (i, j) \in \boldsymbol{G}$.
d_{ij}	travel distance on the link from node i to node j, $\forall (i, j) \in \boldsymbol{G}$, km
$rcap_{ij}$	capacity of each road link (i, j), $\forall (i, j) \in \boldsymbol{G}$.
$vcap$	seating capacity of the vehicle, which is the maximum number of passengers that can share a ride.
c_r	AT price, euros/time step.
c_f	fuel cost, euros/km.
c_v	vehicle depreciation cost, euros/day.
c_p	penalty cost if a travel request is rejected by the system, euros/request.
c_d	penalty cost for delivery delay, euros/time step.
μ	expansion coefficient, representing the number of taxis with the same characteristics.
Decision variables	
P^{evt}	binary variable equal to 1 if travel request e is done by vehicle v starting (pick-up) at time instant t, otherwise 0, $\forall e \in \boldsymbol{E}$, $\forall v \in \boldsymbol{V}$, $\forall t \in \boldsymbol{T}$, $a^e \leq t \leq a^e + w^e$.
A^{evt}	binary variable equal to 1 if travel request e is done by vehicle v ending (drop-off) at time instant t, otherwise 0, $\forall e \in \boldsymbol{E}$, $\forall v \in \boldsymbol{V}$, $\forall t \in \boldsymbol{T}$, $a^e + opt^e \leq t \leq a^e + w^e + lon^e$.
$x_{i_{t_1} j_{t_2}}^v$	binary variable equal to 1 if vehicle v drives from i to j from time instant t_1 to t_2, otherwise 0, $\forall (i_{t_1}, j_{t_2}) \in \boldsymbol{M}$, $\forall v \in \boldsymbol{V}$.
F_{ij}^t	integer variable of the vehicle flow on link (i, j) starting from time instant t, $\forall (i, j) \in \boldsymbol{G}$, $\forall t \in \boldsymbol{T}$, $t < T$.
δ_{ij}^t	integer variable of travel time in time steps when travelling on link (i, j) starting from time instant t, $\forall (i, j) \in \boldsymbol{G}$, $\forall t \in \boldsymbol{T}$, $t < T$.



(a) Network with travel requests (b) Network with dynamic travel times

Fig. 1. An example network with travel requests and dynamic travel times.

destination and the time they would like to depart. Fig. 2 shows the relation between the time components for each request. The departure time for request e can take any value between a^e and $a^e + w^e$, and the arrival time for request e can only happen in the interval between $a^e + opt^e$ and $a^e + w^e + lon^e$, which are defined as the time windows. These two time windows are used to limit the value range of index t when defining the variables P^{evt} and A^{evt}.

3.2. Mathematical model

The optimization model [***OP***] for solving the problem defined above has the following formulation. The objective function is:

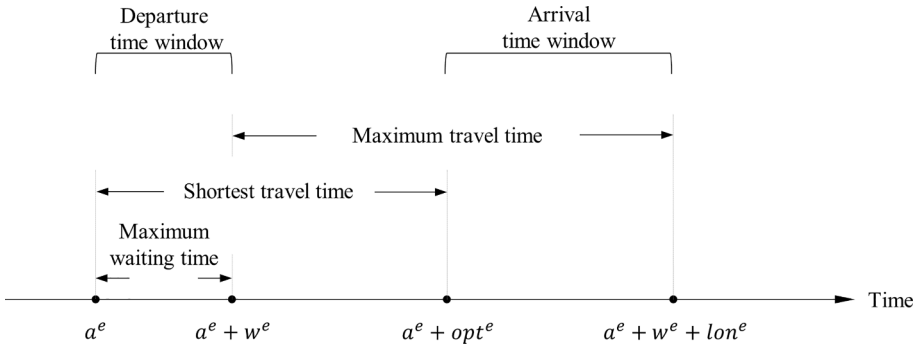


Fig. 2. Time components for each request.

$$[\boldsymbol{OP}] \quad \max Z = \sum_{e\in\boldsymbol{E},v\in\boldsymbol{V},t\in\boldsymbol{T}} c_r\cdot P^{evt}\cdot opt^e - \sum_{\substack{(i_{t_1},j_{t_2})\in\boldsymbol{M}\\ v\in\boldsymbol{V}}} c_f\cdot x^v_{i_{t_1}j_{t_2}}\cdot d_{ij} - c_v\cdot V - \sum_{e\in\boldsymbol{E}} c_p\cdot\left(1 - \sum_{v\in\boldsymbol{V},t\in\boldsymbol{T}} P^{evt}\right) - \sum_{e\in\boldsymbol{E},v\in\boldsymbol{V}} c_d\cdot\left(\sum_{t\in\boldsymbol{T}}(A^{evt}\cdot t) - \sum_{t\in\boldsymbol{T}} P^{evt}\cdot(a^e + opt^e)\right) \tag{1}$$

Subject to:

$$P^{evt} \le \sum_{j_{t_2}\in\{j_{t_2}\,|\,(m^e{}_t,j_{t_2})\in\boldsymbol{M}\}} x^v_{m^e{}_t,j_{t_2}} \qquad \forall\, e\in\boldsymbol{E},\forall\, v\in\boldsymbol{V},\forall\, t\in\boldsymbol{T} \tag{2}$$

$$A^{evt} \le \sum_{i_{t_1}\in\{i_{t_1}\,|\,(i_{t_1},n^e{}_t)\in\boldsymbol{M}\}} x^v_{i_{t_1},n^e{}_t} \qquad \forall\, e\in\boldsymbol{E},\forall\, v\in\boldsymbol{V},\forall\, t\in\boldsymbol{T} \tag{3}$$

$$\sum_{t\in\boldsymbol{T}} P^{evt} = \sum_{t\in\boldsymbol{T}} A^{evt} \qquad \forall\, e\in\boldsymbol{E},\forall\, v\in\boldsymbol{V} \tag{4}$$

$$\sum_{t\in\boldsymbol{T}}(A^{evt}\cdot t) - \sum_{t\in\boldsymbol{T}}(P^{evt}\cdot t) \ge opt^e\cdot\sum_{t\in\boldsymbol{T}} P^{evt} \qquad \forall\, e\in\boldsymbol{E},\forall\, v\in\boldsymbol{V} \tag{5}$$

$$\sum_{t\in\boldsymbol{T}}(A^{evt}\cdot t) - \sum_{t\in\boldsymbol{T}}(P^{evt}\cdot t) \le lon^e\cdot\sum_{t\in\boldsymbol{T}} P^{evt} \qquad \forall\, e\in\boldsymbol{E},\forall\, v\in\boldsymbol{V} \tag{6}$$

$$\sum_{v\in\boldsymbol{V},t\in\boldsymbol{T}} P^{evt} \le 1 \qquad \forall\, e\in\boldsymbol{E} \tag{7}$$

$$\sum_{e\in\boldsymbol{E},t_1\in\boldsymbol{T},t_1\le t} P^{evt_1} - \sum_{e\in\boldsymbol{E},t_2\in\boldsymbol{T},t_2\le t} A^{evt_2} \le vcap \qquad \forall\, t\in\boldsymbol{T},\, t<T,\forall\, v\in\boldsymbol{V} \tag{8}$$

$$\sum_{\substack{(i_{t_1},j_{t_2})\in\boldsymbol{M}\\ t_1\le t,t_2>t}} x^v_{i_{t_1}j_{t_2}} = 1 \qquad \forall\, v\in\boldsymbol{V},\forall\, t\in\boldsymbol{T} \tag{9}$$

$$\sum_{l_{t_2}\in\{l_{t_2}\,|\,(l_{t_2},i_{t_1})\in\boldsymbol{M}\}} x^v_{l_{t_2},i_{t_1}} = \sum_{j_{t_3}\in\{j_{t_3}\,|\,(i_{t_1},j_{t_3})\in\boldsymbol{M}\}} x^v_{i_{t_1}j_{t_3}} \qquad \forall\, i\in\boldsymbol{I},\forall\, t_1\in\boldsymbol{T},\forall\, v\in\boldsymbol{V} \tag{10}$$

$$\sum_{\substack{j_{t_2}\in\{j_{t_2}\,|\,(i_0,j_{t_2})\in\boldsymbol{M}\}\\ v\in\boldsymbol{V}}} x^v_{i_0j_{t_2}} = V \qquad i = initial\ \ station \tag{11}$$

$$F^{t_1}_{ij} = \mu\cdot\sum_{\substack{t_2\in\{t_2\,|\,(i_{t_1},j_{t_2})\in\boldsymbol{M}\}\\ v\in\boldsymbol{V}}} x^v_{i_{t_1}j_{t_2}} \qquad \forall\,(i,j)\in\boldsymbol{G},\forall\, t_1\in\boldsymbol{T},\, t_1<T \tag{12}$$

$$F^t_{ij} \le rcap_{ij} \qquad \forall\,(i,j)\in\boldsymbol{G},\forall\, t\in\boldsymbol{T},\, t<T \tag{13}$$

$$\delta^t_{ij} = \delta^{min}_{ij} + (\delta^{max}_{ij} - \delta^{min}_{ij})\cdot\left(\frac{F^t_{ij}}{rcap_{ij}}\right)^4 \qquad \forall\,(i,j)\in\boldsymbol{G},\forall\, t\in\boldsymbol{T},\, t<T \tag{14}$$

$$\delta^{t_1}_{ij} \le (t_2 - t_1)\cdot x^v_{i_{t_1}j_{t_2}} + \delta^{max}_{ij}\cdot(1 - x^v_{i_{t_1}j_{t_2}}) \qquad \forall\,(i_{t_1},j_{t_2})\in\boldsymbol{M},\forall\, v\in\boldsymbol{V} \tag{15}$$

$$\delta_{ij}^{t_1} \geq (t_2 - t_1)\cdot x_{i_{t_1}j_{t_2}}^{v} + \delta_{ij}^{min}\cdot(1 - x_{i_{t_1}j_{t_2}}^{v}) \quad \forall\ (i_{t_1}, j_{t_2}) \in \boldsymbol{M}, \forall\ v \in \boldsymbol{V} \tag{16}$$

$$t_1 + \delta_{ij}^{t_1} \leq t_2 + \delta_{ij}^{t_2} \quad \forall\ (i, j) \in \boldsymbol{G}, t_1, t_2 \in \boldsymbol{T}, t_1, t_2 < T, t_1 < t_2 \tag{17}$$

$$P^{evt} \in \{0, 1\} \quad \forall\ e \in \boldsymbol{E}, \forall\ v \in \boldsymbol{V}, \forall\ t \in \boldsymbol{T}, a^e \leq t \leq a^e + w^e \tag{18}$$

$$A^{evt} \in \{0, 1\} \quad \forall\ e \in \boldsymbol{E}, \forall\ v \in \boldsymbol{V}, \forall\ t \in \boldsymbol{T}, a^e + opt^e \leq t \leq a^e + w^e + lon^e \tag{19}$$

$$x_{i_{t_1}j_{t_2}}^{v} \in \{0, 1\} \quad \forall\ (i_{t_1}, j_{t_2}) \in \boldsymbol{M}, \forall\ v \in \boldsymbol{V} \tag{20}$$

$$F_{ij}^{t} \in \boldsymbol{N}^0 \quad \forall\ (i, j) \in \boldsymbol{G}, \forall\ t \in \boldsymbol{T}, t < T \tag{21}$$

$$\delta_{ij}^{t} \in \boldsymbol{N}^0 \quad \forall\ (i, j) \in \boldsymbol{G}, \forall\ t \in \boldsymbol{T}, t < T \tag{22}$$

The objective function (1) is considered from both the AT company and the passengers' perspectives. It is a generalized cost-benefit summation for the two components of the system. For the AT company, it aims to maximize the daily profit including the revenue from the AT clients, the vehicle fuel costs and the vehicle depreciation costs. The revenue is only charged based on the shortest path of each request, if that was not the case the model would try to detour passengers to charge them for more travel distance and time. The vehicle depreciation cost is considered based on the number of vehicles, which means that this cost is a constant component in the objective function and will not influence the solution space of the problem. However, we decide to keep the depreciation cost since we want to analyse the monetary impact of the fleet size on the system profit. For the passengers who get rejected by the system, a penalty is paid as referred above. The delivery delay is also penalized in order to internalize the level of service offered to the customers. This consists of late departure (waiting time) and the congestion delay in relation to the shortest duration path.

Constraints (2) and (3) impose that request e from its origin to node j (from node i to its destination) can only be satisfied by vehicle v at time instant t if that vehicle has passed through the origin (destination) at the same time instant. Constraints (4) assure that the satisfied travel request must have a departure time and an arrival time. Constraints (5) and (6) impose that the travel time must be between the shortest and the longest travel time of that request. The shortest travel time opt^e of each request e is calculated by the shortest path method using the minimum link travel time δ_{ij}^{min} which means that any feasible path in this time-space network will satisfy constraints (5). However, we did a comparison test in programming with and without constraints (5) and the results show that the model with constraints (5) achieve the same solution with a shorter computing time (they act as valid inequalities for the problem). This means that constraints (5) do not change the solution space but they help cut the space of this optimization problem and accelerate the searching process in the branch and bound algorithm. Constraints (7) ensure that a travel request can only be served by one vehicle. Constraints (8) impose that the number of passengers loaded on each vehicle during time step t to $t + 1$ cannot exceed the vehicle's seating capacity. The left-hand side of this set of constraints computes the number of passengers on board from time instant t to $t + 1$ by summing all the pick-up and drop-off activities from the beginning to time instant t. When the AT's seating capacity is larger than 1, it is possible to have ride-sharing. Constraints (9) ensure that at each time instant each vehicle must have one status: starting from one node to another node or be in the middle of a link. Constraints (10) are the flow conservation constraints which make sure that the number of taxis leaving from node i from time instant t is equal to the number of vehicles arriving at node i at time instant t. Constraints (11) describe the initial status of the AT fleet. Constraints (12) compute the flow of vehicles on each road link (i, j) from time instant t_1. Sampling expansion coefficient is used to let each AT represent μ real ATs following the same concept proposed by (Correia and van Arem, 2016a). Constraints (13) limit the flow by the capacity of each link. Constraints (14) compute the dynamic travel time of each road link which is a function of the AT flow on that link. In this paper, we consider the travel time given by a BPR function (Dafermos and Sparrow, 1969): $t(V) = t_0\left(1 + a \times \left(\frac{V}{Q}\right)^b\right)$, where $t(V)$ is the travel time when the traffic flow is V, t_0 is the zero-flow travel time; V is the traffic flow; Q is the road capacity; a and b are estimation parameters. The travel time is an integer number of time steps. Constraints (15)–(16) guarantee that the travel time computed from constraints (14) is imposed to the decision variable $x_{i_{t_1}j_{t_2}}^{v}$. Constraints (17) guarantee the first in first out (FIFO) conditions. We build the assumption that vehicles entering in a link (i, j) later from node i should not leave earlier from node j, which means the ATs do not pass on another. These constraints were established by Kaufman et al. (1992). Constraints (18)–(22) define the domain for the decision variables.

4. Rolling horizon

Model [***OP***] described in Section 3 is a static DARP problem assuming all demand given. However even if all demand were known in advance, it would hardly be solvable for a whole day due to its complexity. In this section, we present the rolling horizon framework that allows for real-time requests.

4.1. Framework setting

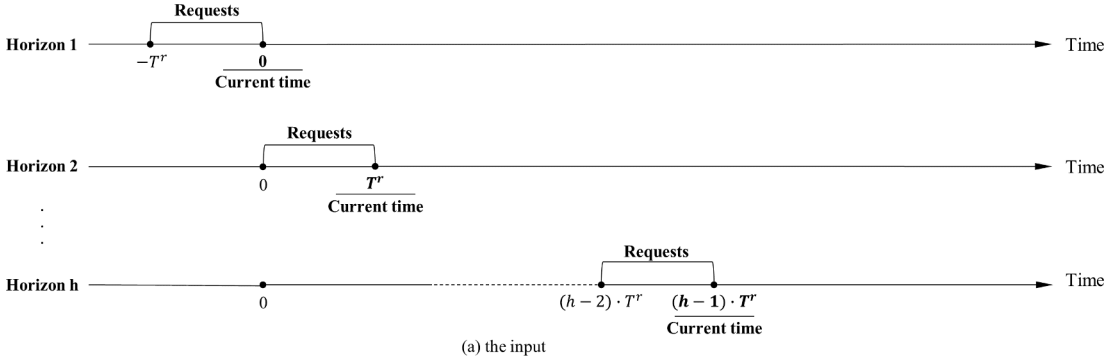
The notations used in this section are presented in Table 2.

We introduce a rolling horizon framework to divide the whole optimization period into several horizons to address the real-time demand. This framework uses model [***OP***] in every horizon with the real-time demand to obtain the AT routes. After that, the

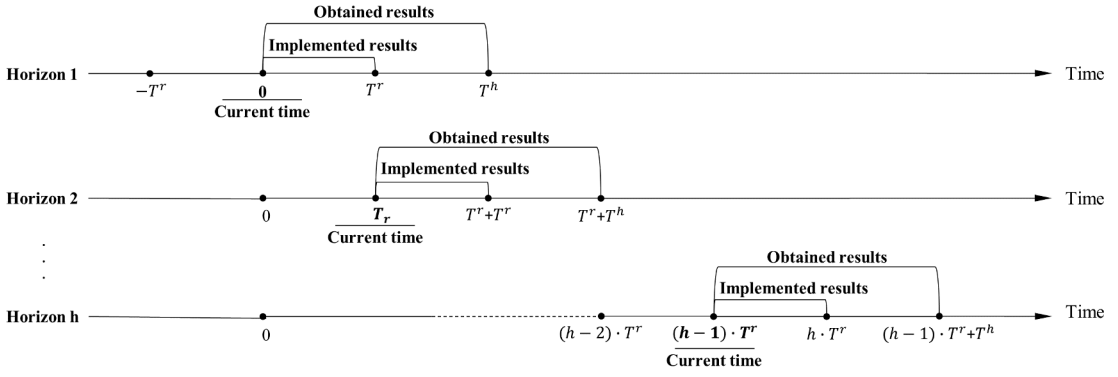
Table 2
Notations.

Notation	Description
Sets	
$\boldsymbol{H}$	$= \{1, \cdots, h, \cdots H\}$, set of horizons in an operation day, where H is the total number of horizons.
$\boldsymbol{E'}$	set of the requests which are partially implemented from the previous horizon.
$\boldsymbol{E^h}$	set of the requests which belong to horizon h, $\forall\, h \in \boldsymbol{H}$.
$\boldsymbol{T^h}$	$= \{0, \cdots, t, \cdots, T^h\}$, set of time instants in a horizon, where T^h replaces T in the model (1)–(22).
Parameters	
T^r	time length for rolling, $T^r < T^h$.
loc^v	location of vehicle v at or closest to the end of the implemented period, $\forall\, v \in \boldsymbol{V}$.
int^v	time instant when vehicle v is available in next horizon, $\forall\, v \in \boldsymbol{V}$.
veh^e	the vehicle that satisfies request e in the current horizon, $\forall\, e \in \boldsymbol{E^h}$, $\forall\, h \in \boldsymbol{H}$.
st_{it}^v	equals to 1 if vehicle v is travelling on a link which will finish in next horizon from time instant t to $t+1$, otherwise 0, $\forall\, v \in \boldsymbol{V}$, $\forall\, t \in \boldsymbol{T^h}$, $t < T^h$, $i = loc^v$.
sg_{it}^v	equals to 1 if vehicle v is travelling on a link which will finish in next horizon from time instant t to $t+1$ and will end this trip at time instant $t+1$, otherwise 0, $\forall\, v \in \boldsymbol{V}$, $\forall\, t \in \boldsymbol{T^h}$, $t < T^h$, $i = loc^v$.
$\tilde{x}_{i_{t_1}j_{t_2}}^v$	value of the variable $x_{i_{t_1}j_{t_2}}^v$, $\forall\, (i_{t_1}, j_{t_2}) \in \boldsymbol{M}$, $\forall\, v \in \boldsymbol{V}$.
$\tilde{P}^{evt}$	value of the variable P^{evt}, $\forall\, e \in \boldsymbol{E^h}$, $\forall\, v \in \boldsymbol{V}$, $\forall\, t \in \boldsymbol{T^h}$, $\forall\, h \in \boldsymbol{H}$.
$\tilde{A}^{evt}$	value of the variable A^{evt}, $\forall\, e \in \boldsymbol{E^h}$, $\forall\, v \in \boldsymbol{V}$, $\forall\, t \in \boldsymbol{T^h}$, $\forall\, h \in \boldsymbol{H}$.

optimization horizon rolls forward with a specific rolling length and reaches the next horizon with updated AT requests. The rolling horizon simplifies an original problem where all demand is taken as given but it is also a valid and effective framework when the travel demand information comes in real-time. When we want to solve the problem with real-time travel requests, the rolling horizon is not seen as a heuristic approach. It is rather a framework to address AT's DARP when the input will be updated during the implementation of the route. Nevertheless, we agree that this simplifies the solving of the mathematical problem as far as the number of variables and constraints are concerned.



(a) the input



(b) the output

Fig. 3. Rolling horizon framework.

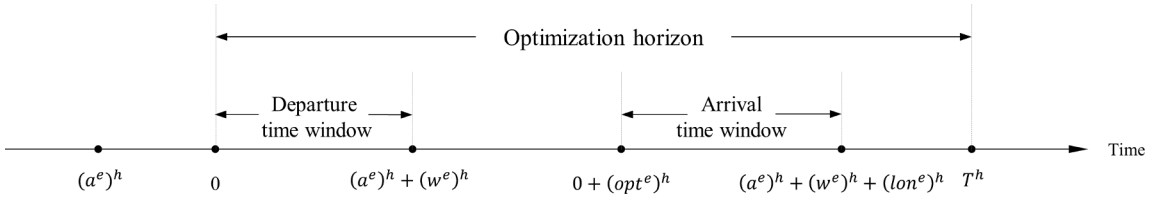


Fig. 4. Departure and arrival time windows for the requests.

Fig. 3 shows the rolling horizon framework. Time instant 0 is defined as the time where the ATs start to move. For horizon 1, we run the optimization model [***OP***] with the requests that occurred in the past which have the desired departure time from $-T^r$ to 0. Once the optimal solutions are obtained, the AT system will partially implement the routing results from 0 to T^r. Then the optimization horizon rolls forward with the time length T^r and arrives at the next horizon. The running process will be done just before the start time instant of each horizon. When it rolls to horizon h, the demand will include the requests happening from $(h-2)\cdot T^r$ to $(h-1)\cdot T^r$ and the implemented time period will be $(h-1)\cdot T^r$ to $h\cdot T^r$

The horizon and rolling length will affect the performance of the optimization: the horizon length determines how far the system will make a plan for the routing of ATs and the rolling length determines how often the system will input the new requests, update the optimal results and implement the routing plan. We set the rolling length shorter than the horizon length because the system can plan for the current requests, implement a part of the routing plan and leave the other part of the plan to be updated with the new entering requests. If the horizon is the same as the rolling length, the system will implement exactly what is calculated from the model and have no chance to modify the route for the following requests.

The updated time windows for the requests can be seen in Fig. 4. Firstly, the desired departure time for the requests which will be analysed in horizon h should be transformed from an absolute time to a relative time in that specific horizon by Eq. (23). The time they are submitted is the desired departure time, which is a negative value related to the current horizon h. Since these requests can only be considered in the next horizon after they are submitted, the earliest departure time is the start time of the next horizon. The departure and arrival time window for these requests can be calculated by Eqs. (24) and (25).

$$(a^e)^h = a^e - (h-1)\cdot T^r, \quad \forall\, e \in \boldsymbol{E^h}, \forall\, h \in \boldsymbol{H} \tag{23}$$

$$0 \le t \le (a^e)^h + (w^e)^h, \quad \forall\, e \in \boldsymbol{E^h}, \forall\, h \in \boldsymbol{H} \tag{24}$$

$$opt^e \le t \le (a^e)^h + (w^e)^h + (lon^e)^h, \forall\, e \in \boldsymbol{E^h}, \forall\, h \in \boldsymbol{H} \tag{25}$$

4.2. The updated model

If model [***OP***] is applied to the first horizon then we are able to calculate the following parameters' values, which are crucial in guaranteeing model's continuity.

$$loc^v = \sum_{\substack{(i_{t_1}j_{t_2})\in \boldsymbol{M} \\ t_1<T^r, t_2\ge T^r}} \tilde{x}^v_{i_{t_1}j_{t_2}} \cdot j \quad \forall\, v \in \boldsymbol{V} \tag{26}$$

$$int^v = \sum_{\substack{(i_{t_1}j_{t_2})\in \boldsymbol{M} \\ t_1<T^r, t_2\ge T^r}} \tilde{x}^v_{i_{t_1}j_{t_2}} \cdot (t_2 - T^r) \quad \forall\, v \in \boldsymbol{V} \tag{27}$$

$$veh^e = \sum_{v\in \boldsymbol{V}, t\in \boldsymbol{T^h}} \tilde{P}^{evt} \cdot v \quad \forall\, e \in \boldsymbol{E^h} \tag{28}$$

Eq. (26) calculate the final location of vehicle v at the end of the implementation period T^r. If this vehicle is in the middle of a link, then the location will be the destination of the implemented period. Eq. (27) equal to 0 if vehicle v ends its trip and becomes available at the end of the current horizon. If not, it will be the first available time instant when vehicle v ends travelling its link in the next horizon. Eq. (28) indicate for travel request e which vehicle satisfies it. The values of $st^v_{i_t}$ are obtained as follows: if the first available time of vehicle v is later than the beginning of the next horizon ($int^v > 0$), then the value of $st^v_{i_t}$ equals to 1, for $i = loc^v$; otherwise 0. If t to $t+1$ is the last time step vehicle v is finishing travelling on a link and it will be released at $t+1$, then the value of $sg^v_{i_t}$ equals to 1; otherwise 0.

Based on model [***OP***] and the system status of the previous horizon given by (26)–(28), the updated model implemented in horizon h under the rolling horizon framework [$\boldsymbol{OP^h}$] is defined as follows:

$$[\boldsymbol{OP^h}] \quad \max Z^h = \sum_{e\in \boldsymbol{E},v\in \boldsymbol{V},t\in \boldsymbol{T^h}} c_r\cdot P^{evt}\cdot opt^e - \sum_{\substack{(i_{t_1}j_{t_2})\in \boldsymbol{M}\\ v\in \boldsymbol{V}}} c_f\cdot x^v_{i_{t_1}j_{t_2}}\cdot d_{ij} - c_v\cdot V - \sum_{e\in \boldsymbol{E^h}} c_p\cdot\left(1 - \sum_{v\in \boldsymbol{V},t\in \boldsymbol{T^h}} P^{evt}\right)$$
$$- \sum_{e\in \boldsymbol{E^h},v\in \boldsymbol{V}} c_d\cdot\left(\sum_{t\in \boldsymbol{T^h}} (A^{evt}\cdot t) - \sum_{t\in \boldsymbol{T^h}} P^{evt}\cdot((a^e)^h + (opt^e)^h)\right) \tag{29}$$

Subject to:

(2)–(8), (13)–(22) plus

$$\sum_{\substack{(i_{t_1}j_{t_2})\in \boldsymbol{M}\\ t_1\le t, t_2>t}} x^v_{i_{t_1}j_{t_2}} + \sum_{i\in \boldsymbol{N}} st^v_{i_t} = 1 \quad \forall\, v\in \boldsymbol{V}, \forall\, t\in \boldsymbol{T^h} \tag{30}$$

$$\sum_{l_{t_1}\in\{l_{t_1}|(l_{t_1},i_t)\in \boldsymbol{M}\}} x^v_{l_{t_1}i_t} + sg^v_{i_{t-1}} = \sum_{j_{t_2}\in\{j_{t_2}|(i_t,j_{t_2})\in \boldsymbol{M}\}} x^v_{i_tj_{t_2}} \quad \forall\, i\in \boldsymbol{I}, \forall\, t\in \boldsymbol{T^h}, \forall\, v\in \boldsymbol{V} \tag{31}$$

$$\sum_{j_{t_2}\in\{j_{t_2}|(i_{t_1}j_{t_2})\in \boldsymbol{M}\}} x^v_{i_{t_1}j_{t_2}} = 1 \quad \forall\, v\in \boldsymbol{V}, i = (loc^v)^h, t_1 = (int^v)^h \tag{32}$$

$$F_{ij} = \mu\cdot \sum_{\substack{(t_1,t_2)\in \boldsymbol{T^h}\\ v\in \boldsymbol{V}}} x^v_{i_{t_1}j_{t_2}} \quad \forall\,(i,j)\in \boldsymbol{G} \tag{33}$$

$$P^{evt} = 1 \quad \forall\, e\in \boldsymbol{E}', v = (veh^e)^h, t = (int^v)^h \tag{34}$$

Constraints (30) are modified from constraints (9), indicating that one AT can only have one status out of two: driving on a link from the current horizon, or driving on a link which has not been finished in the previous horizon. Constraints (31) are an update of constraints (10). Vehicles arriving at i_t are not only from the trips in the current horizon, but also from the previous one. Constraints (32) impose that all vehicles must start from the same location in which they have stayed in the previous horizon, which replace constraints (11). Constraints (33) compute the link flow as the total number of ATs travelling on link (i, j) within one horizon, which is an update of constraints (12). In this section, we extend the time scale of the link flow and calculate it in a cumulative way. As a result, the index t of the variables F^t_{ij} and t^t_{ij} is eliminated, which also reduces the number of variables in $[\boldsymbol{OP^h}]$. Constraints (34) guarantee that the partially-implemented requests from the previous horizon must be served continuously.

The following pseudo-code shows the solving process under the rolling horizon framework.

Step 0: Initialize the locations of ATs
  set $Label(e) = 0, \forall\, e\in \boldsymbol{E^h}, h = 1$
Step 1: Filter the demand for horizon h
  $\forall\, e\in \boldsymbol{E^h}$, $Label(e) = 0$
    If $(h-2)\cdot T^r \le (a^e)^h < (h-1)\cdot T^r$, then $Label(e) = 1$
    end-if
Step 2: Run model $[\boldsymbol{OP^h}]$ with objective function (29),
  subject to (2)–(8), (13)–(22), (30)–(33), $\forall\, e\in \boldsymbol{E^h}$, $Label(e) > 0$ and (34), $\forall\, e\in \boldsymbol{E^h}$, $Label(e) = 2$
Step 3: Save the vehicle routing information according to (26) and (27)
Step 4: Save the request satisfying information
  $\forall\, e\in \boldsymbol{E^h}$, $Label(e) > 0$
    Step 4.0: If $\sum_{t,v}(\tilde{P}^{evt}\cdot t) < T^r$ and $\sum_{t,v}(\tilde{A}^{evt}\cdot t) \le T^r$, then go to Step 4.1
    else if $\sum_{t,v}(\tilde{P}^{evt}\cdot t) < T^r$ and $\sum_{t,v}(\tilde{A}^{evt}\cdot t) > T^r$, then go to Step 4.2
      else if $\sum_{t,v}(\tilde{P}^{evt}\cdot t) \ge T^r$, then go to Step 4.3.
    end-if
    Step 4.1: Save the values of $\sum_{t,v}(\tilde{P}^{evt}\cdot t)$ and $\sum_{t,v}(\tilde{A}^{evt}\cdot t)$ as the final departure and arrival time of request e
      set $Label(e) = -1$
    $e = e + 1$, go to Step 4.0
    Step 4.2: Save the continuity information of each partially implemented request e
      Step 4.2.1: Save the AT's number who serves this request $\tilde{v} = veh^e$ according to (28)
        set the new origin of this request $(m^e)^{h+1} = loc^{\tilde{v}}$
          set the new time schedule of this request $(a^e)^{h+1} = int^{\tilde{v}}$, $(w^e)^{h+1} = 0$
      Step 4.2.2: Set $Label(e) = 2$
      $e = e + 1$, go to Step 4.0
    Step 4.3: Set $Label(e) = 0$
      $e = e + 1$, go to Step 4.0
Step 5: If $h = H$ then finish
  otherwise, $h = h + 1$, go to Step 1

---

5. Solution algorithm

The INLP model [$\boldsymbol{OP^h}$] can be solved using most of the commercial software like Xpress or CPLEX. But generally, the computation time is excessively long, especially when applying it to large scale problems, due to its NP-hard property. To solve a large-scale problem as described in Sections 3 and 4, an efficient solution algorithm is needed. In this paper, we develop a customized Lagrangian relaxation algorithm (Fisher, 1981) to solve model [$\boldsymbol{OP^h}$].

Fisher (2004, 1981) showed that this approach can efficiently solve a wide range of difficult mixed integer problems, e.g. the travelling salesman problem, the scheduling problem, the location problem, the assignment problem, etc. An et al. (2017) applied it to solve a sensor location problem for object positioning and surveillance, where the Lagrangian relaxation provides a lower bound (minimization problem) to the original problem. Bai et al. (2011) introduced a Lagrangian relaxation-based heuristic algorithm to solve the biofuel refinery location problem under traffic congestion and obtain the near-optimal feasible solutions efficiently. Lei and Ouyang (2018) proposed a Lagrangian relaxation-based algorithm to solve the one-commodity pick-up and delivery problem. The numerical experiments show that it is able to generate a good solution for large-scale cases in short computation time. The pick-up and delivery problem was also addressed by Imai et al. (2007) in the field of container load from/to an intermodal terminal and solved by a sub-gradient heuristic based on a Lagrangian relaxation to identify a near-optimal solution. Shen et al. (2011) studied an inventory routing problem in crude oil transportation and developed a Lagrangian relaxation approach for finding the near-optimal solution of the mixed integer problem. All these applications demonstrate that this algorithm can be used to provide bounds in discrete optimization and can be further integrated with other methods e.g. branch-and-bound, heuristics to produce a near-optimal solution to these challenging problems.

5.1. Lagrangian relaxation

The notations used in this section are presented in Table 3.

In [$\boldsymbol{OP^h}$], the vehicle routing variables $x^v_{i_{t_1}j_{t_2}}$ are related to the passenger assignment variables P^{evt} and A^{evt} by constraints (2) and (3). Such a relation makes the model complicated and computationally challenging. To decouple them, we relax constraints (2) and (3) and incorporate them into the objective function (29) with nonnegative Lagrangian multipliers μ_1^{evt} and μ_2^{evt}. The relaxed problem can be written as follows:

$$
\begin{aligned}
[\boldsymbol{RP^h}] \quad \max Z' = & \sum_{e\in \boldsymbol{E^h}, v\in \boldsymbol{V}, t\in \boldsymbol{T^h}} c_r \cdot P^{evt} \cdot opt^e - \sum_{\substack{(t_1,t_2)\in \boldsymbol{M} \\ v\in \boldsymbol{V}}} c_f \cdot x^v_{i_{t_1}j_{t_2}} \cdot d_{ij} - c_v \cdot V - \sum_{e\in \boldsymbol{E^h}} c_p \cdot \left(1 - \sum_{v\in \boldsymbol{V}, t\in \boldsymbol{T^h}} P^{evt}\right) \\
& - \sum_{e\in \boldsymbol{E^h}, v\in \boldsymbol{V}} c_d \cdot \left(\sum_{t\in \boldsymbol{T^h}} (A^{evt}\cdot t) - \sum_{t\in \boldsymbol{T^h}} P^{evt}\cdot((a^e)^h + (w^e)^h)\right) - \sum_{e\in \boldsymbol{E^h}, v\in \boldsymbol{V}, t\in \boldsymbol{T^h}} \mu_1^{evt} \cdot \left(P^{evt} - \sum_{j_{t_2}\in\{j_{t_2}\,|\,(m^e_t, j_{t_2})\in \boldsymbol{M}\}} x^v_{m^e_t, j_{t_2}}\right) \\
& - \sum_{e\in \boldsymbol{E^h}, v\in \boldsymbol{V}, t\in \boldsymbol{T^h}} \mu_2^{evt} \cdot \left(A^{evt} - \sum_{i_{t_1}\in\{i_{t_1}\,|\,(i_{t_1}, n^e_t)\in \boldsymbol{M}\}} x^v_{i_{t_1}, n^e_t}\right)
\end{aligned} \tag{35}
$$

subject to (4)–(8), (13)–(22) and (30)–(34).

For given μ_1^{evt} and μ_2^{evt}, the optimal solution of [$\boldsymbol{RP^h}$] provides an upper bound to the original problem [$\boldsymbol{OP^h}$]. The [$\boldsymbol{RP^h}$] can be further decomposed into two sub-problems: the passenger assignment problem [$\boldsymbol{SP}1$] and the vehicle routing problem [$\boldsymbol{SP}2$].

The passenger assignment problem involving variables P^{evt} and A^{evt} becomes the following:

Table 3
Notations.

Notation	Description
Sets	
$\boldsymbol{L}$	$= \{1, \cdots, l, \cdots L\}$, set of Lagrangian iterations, where L is the total number of iterations.
$\boldsymbol{K}$	$= \{0, \cdots, k, \cdots K\}$, set of traffic assignment iterations, where K is the total number of iterations.
Parameters	
μ_1^{evt}	Lagrangian multiplier associated with constraints (2), $\forall\, e \in \boldsymbol{E^h}$, $\forall\, v \in \boldsymbol{V}$, $\forall\, t \in \boldsymbol{T^h}$, $\forall\, h \in \boldsymbol{H}$.
μ_2^{evt}	Lagrangian multiplier associated with constraints (3), $\forall\, e \in \boldsymbol{E^h}$, $\forall\, v \in \boldsymbol{V}$, $\forall\, t \in \boldsymbol{T^h}$, $\forall\, h \in \boldsymbol{H}$.
ρ^l	step size of Lagrangian relaxation, $\forall\, l \in \boldsymbol{L}$.
UB^l	current upper bound obtained in iteration l, $\forall\, l \in \boldsymbol{L}$.
LB^*	best lower bound found.
π^l	control parameter, $\forall\, l \in \boldsymbol{L}$.
F_{ij}	value of the variable F_{ij}, $\forall\, (i,j) \in \boldsymbol{G}$.
δ_{ij}	value of travel times in time steps when travelling on link (i,j), $\forall\, (i,j) \in \boldsymbol{G}$.
Vol_{ij}	value of ATs' volumes when travelling on link (i,j), $\forall\, (i,j) \in \boldsymbol{G}$.

$$[\boldsymbol{SP}1] \quad \max Z_1 = \sum_{e\in \boldsymbol{E^h}, v\in \boldsymbol{V}, t\in \boldsymbol{T^h}} c_r \cdot P^{evt} \cdot opt^e - \sum_{e\in \boldsymbol{E^h}} c_p \cdot \left(1 - \sum_{v\in \boldsymbol{V}, t\in \boldsymbol{T^h}} P^{evt}\right) - \sum_{e\in \boldsymbol{E^h}, v\in \boldsymbol{V}} c_d \cdot \left(\sum_{t\in \boldsymbol{T^h}} (A^{evt}\cdot t) - \sum_{t\in \boldsymbol{T^h}} P^{evt}\cdot((a^e)^h + (w^e)^h)\right) - \sum_{e\in \boldsymbol{E^h}, v\in \boldsymbol{V}, t\in \boldsymbol{T^h}} \mu_1^{evt} \cdot P^{evt} - \sum_{e\in \boldsymbol{E^h}, v\in \boldsymbol{V}, t\in \boldsymbol{T^h}} \mu_2^{evt} \cdot A^{evt} \tag{36}$$

Subject to (4)–(8), (18), (19) and (34).

This is a linear programming formulation with binary variables P^{evt} and A^{evt}. It can be solved directly by a commercial solver.

Meanwhile, the vehicle routing problem involving variables $x^v_{i_{t_1}j_{t_2}}$, F_{ij} and δ_{ij} considering traffic congestion can be written as follows:

$$[\boldsymbol{SP}2] \quad \max Z_2 = -\sum_{\substack{(t_1,t_2)\in \boldsymbol{M} \\ v\in \boldsymbol{V}}} c_f \cdot x^v_{i_{t_1}j_{t_2}} \cdot d_{ij} - c_v \cdot V + \sum_{e\in \boldsymbol{E^h}, v\in \boldsymbol{V}, t\in \boldsymbol{T^h}} \mu_1^{evt} \cdot \sum_{j_{t_2}\in\{j_{t_2}\,|\,(m^e{}_t, j_{t_2})\in \boldsymbol{M}\}} x^v_{m^e{}_t j_{t_2}} + \sum_{e\in \boldsymbol{E^h}, v\in \boldsymbol{V}, t\in \boldsymbol{T^h}} \mu_2^{evt} \cdot \sum_{i_{t_1}\in\{i_{t_1}\,|\,(i_{t_1}, n^e{}_t)\in \boldsymbol{M}\}} x^v_{i_{t_1} n^e{}_t} \tag{37}$$

Subject to (13)–(17), (20)–(22) and (30)–(33).

Sub-problem [$\boldsymbol{SP}2$] is a non-linear optimization model due to the travel time constraints (14). Moreover, routing all the vehicles with dynamic travel time generates a great number of decision variables and constraints, which makes it challenging to solve. We propose an iterative assignment process to exclude the non-linear constraints (14) and update the link travel time based on the traffic flow that results from the optimization process, following a similar concept proposed by (Correia and van Arem, 2016a). We have mentioned in Section 3.1 that we do not allow parking in this model, otherwise, all ATs would stay at the initial nodes and no routing would happen. This is because in [$\boldsymbol{SP}2$] there are no demand nodes and ATs choose paths only according to the impedance of each link, which will vary with the values of the Lagrangian multipliers in several iterations. The iterative assignment process is conducted as follows:

a. Compute the initial travel times, i.e. the minimum travel time on each link as the input travel times.
b. Design the routing of each AT based on the input travel times by solving [$\boldsymbol{SP}2$] with objective function (37) and constraints (13), (20)–(21) and (30)–(33).
c. Calculate the traffic flow on each link according to the optimal results from step b.
d. Update the travel times according to the BPR function based on the traffic flows from step c.
e. A set of errors is computed between the updated travel times and the input travel times used in [$\boldsymbol{SP}2$].
f. If all the errors meet the stopping criterion, then the current solution is the final solution of [$\boldsymbol{SP}2$]; otherwise, go back to step b using the updated travel times as input travel times.

In this process, the travel times do no change within the optimization model, which decreases the number of vehicle movement variables $x^v_{i_{t_1}j_{t_2}}$ significantly. Additionally, without constraints (14), [$\boldsymbol{SP}2$] becomes a linear formulation and it is easy to solve in each iteration.

5.2. Upper bound and lower bound

The upper bound represents the possible best objective function value of [$\boldsymbol{OP^h}$], which means that we will never find a feasible solution better than that one. The lower bound is the best feasible solution that has been obtained for [$\boldsymbol{OP^h}$]. It also indicates that the global optimal solution will not be worse than that. By solving the above two sub-problems, the summation of their objective function values with given μ_1^{evt} and μ_2^{evt} constitutes an upper bound to the [$\boldsymbol{OP^h}$]. However, the optimal solution of [$\boldsymbol{SP}1$] and [$\boldsymbol{SP}2$] may violate the relaxed constraints (2) and (3), which make the solution to [$\boldsymbol{OP^h}$] infeasible. In this case, we propose a semi-optimization method to adjust the infeasible solution to a feasible one, therefore producing a lower bound to [$\boldsymbol{OP^h}$]. In each iteration of the Lagrangian relaxation algorithm, the semi-optimization method has the following steps:

a. Obtain the solution values of AT routing variable $x^v_{i_{t_1}j_{t_2}}$ from [$\boldsymbol{SP}2$] as $\tilde{x}^v_{i_{t_1}j_{t_2}}$.
b. Use the values of $\tilde{x}^v_{i_{t_1}j_{t_2}}$ as an input to [$\boldsymbol{OP^h}$].
c. Solve [$\boldsymbol{OP^h}$] with decision variables on travel request and vehicle matching, i.e. P^{evt} and A^{evt}, objective function (29), and constraints (2)–(8), (18), (19) and (34)
d. Save the optimal solution from step c as a feasible solution to [$\boldsymbol{OP^h}$] and obtain the lower bound of this Lagrangian iteration.

With the values of $\tilde{x}^v_{i_{t_1}j_{t_2}}$ from step a the routings of all the ATs are known including the effect of traffic congestion. Nevertheless, there is no information on which requests are served by these ATs. Then we keep these routing results and use them to match the passengers' requests in the dimensions of space and time by solving the model in step c. If there are some requests satisfied by the ATs, then this is a feasible solution for providing the AT service. This solution satisfies all the constraints in [$\boldsymbol{OP^h}$] hence, providing a lower bound to the [$\boldsymbol{OP^h}$].

After each Lagrangian iteration, the multipliers μ_1^{evt} and μ_2^{evt} will be updated using the standard sub-gradient procedure (Fisher,

1981) as follows:

$$(\mu_1^{evt})^{l+1} = \max\left\{0, (\mu_1^{evt})^l + \rho^l \cdot \left(\tilde{P}^{evt} - \sum_{j_{t2} \in \{j_{t_2} | (m^e{}_t, j_{t_2}) \in \boldsymbol{M}\}} \tilde{x}^v_{m^e{}_t, j_{t_2}}\right)\right\} \quad \forall\, e \in \boldsymbol{E}, \forall\, v \in \boldsymbol{V}, \forall\, t \in \boldsymbol{T^h} \tag{38}$$

$$(\mu_2^{evt})^{l+1} = \max\left\{0, (\mu_2^{evt})^l + \rho^l \cdot \left(\tilde{A}^{evt} - \sum_{i_{t1} \in \{i_{t_1} | (i_{t_1}, n^e{}_t) \in \boldsymbol{M}\}} \tilde{x}^v_{i_{t1}, n^e{}_t}\right)\right\} \quad \forall\, e \in \boldsymbol{E}, \forall\, v \in \boldsymbol{V}, \forall\, t \in \boldsymbol{T^h} \tag{39}$$

$$\rho^l = \frac{\pi^l \cdot (UB^l - LB^*)}{\sum_{e \in \boldsymbol{E}, v \in \boldsymbol{V}, t \in \boldsymbol{T^h}} \left(\tilde{P}^{evt} - \sum_{j_{t_2} \in \{j_{t_2} | (m^e{}_t, j_{t_2}) \in \boldsymbol{M}\}} \tilde{x}^v_{m^e{}_t, j_{t_2}}\right)^2 + \sum_{e \in \boldsymbol{E}, v \in \boldsymbol{V}, t \in \boldsymbol{T^h}} \left(\tilde{A}^{evt} - \sum_{i_{t_1} \in \{i_{t_1} | (i_{t_1}, n^e{}_t) \in \boldsymbol{M}\}} \tilde{x}^v_{i_{t1}, n^e{}_t}\right)^2} \tag{40}$$

where $\tilde{P}^{evt}$, $\tilde{A}^{evt}$ and $\tilde{x}^v_{i_{t1} j_{t_2}}$ are the values of decision variables from [***SP***1] and [***SP***2] in the l^{th} iteration; $\pi^{l=1} = 2$ as an initial value and it will be halved when the upper bound has failed to improve in 3 Lagrangian iterations. The initial values of the Lagrangian multipliers are set as $\mu_1^{evt} = 1.5$, $\mu_2^{evt} = 1.5$, $\forall\, e \in \boldsymbol{E^h}$, $\forall\, v \in \boldsymbol{V}$, $\forall\, t \in \boldsymbol{T^h}$, $\forall\, h \in \boldsymbol{H}$.

The following pseudo-code shows the customized Lagrangian relaxation algorithm, where l is the Lagrangian iteration number, k is the traffic assignment iteration number. This solving process is used to solve [$\boldsymbol{OP^h}$] for horizon h, which should be embedded in the rolling horizon framework. It replaces Step 2 in the pseudo-code presented in Section 4.

Step 0: Initialize $(\mu_1^{evt})^{l=1} = 1.5$, $(\mu_2^{evt})^{l=1} = 1.5$, $\forall\, e \in \boldsymbol{E^h}$, $\forall\, v \in \boldsymbol{V}$, $\forall\, t \in \boldsymbol{T^h}$

Step 1: Solve [***SP***1] with $(\mu_1^{evt})^l$ and $(\mu_2^{evt})^l$;

Obtain the value of $(Z_1)^l$, $\tilde{P}^{evt}$ and $\tilde{A}^{evt}$

Step 2: Solve[***SP***2]

Step 2.0: Initialize $(\tilde{\delta}_{ij})^{k=0} = \delta_{ij}^{min}$, $LB^* = 0$

Step 2.1: Solve [***SP***2] with $(\tilde{\delta}_{ij})^k$ as input values instead of decision variables;

save the value of F_{ij} from variableF_{ij}

Step 2.2: If $k = 0$ then

$$(Vol_{ij})^0 = \tilde{F}_{ij} \qquad \forall\, (i,j) \in \boldsymbol{G}$$

else

$$(Vol_{ij})^k = \left(1 - \frac{1}{K}\right) \cdot (Vol_{ij})^{k-1} + \frac{1}{K} \cdot \tilde{F}_{ij} \qquad \forall\, (i,j) \in \boldsymbol{G}$$

end-if

Step 2.3: Update the link travel times

$$(\tilde{\delta}_{ij})^{k+1} = \delta_{ij}^{min} + (\delta_{ij}^{max} - \delta_{ij}^{min}) \cdot \left(\frac{(Vol_{ij})^k}{rcap_{ij}}\right)^4 \qquad \forall\, (i,j) \in \boldsymbol{G}$$

Step 2.4: If $(\tilde{\delta}_{ij})^{k+1} - (\tilde{\delta}_{ij})^k \le stopping\ \ criterion\ \ 1 \qquad \forall\, (i,j) \in \boldsymbol{G}$ then

save the values of $(Z_2)^l$ and $\tilde{x}^v_{i_{t1} j_{t_2}}$

go to Step 3

otherwise, $k = k + 1$, go to Step 2.1

Step 3: Find the feasible solution

Step 3.1: Put the values of $\tilde{x}^v_{i_{t1} j_{t_2}}$ from Step 2.4 into [$\boldsymbol{OP^h}$]

Step 3.2: Solve[$\boldsymbol{OP^h}$]

Step 3.3: Save the values of variable P^{evt} and A^{evt} together with $\tilde{x}^v_{i_{t1} j_{t_2}}$ from Step 2.4,

then this is a feasible solution to [$\boldsymbol{OP^h}$]

Step 3.4: Obtain the value of $(Z^h)^l$

Step 4: $UB^l = (Z_1)^l + (Z_2)^l$

If $h = 1$ then $LB^* = (Z^h)^l$

else if $LB^* < (Z^h)^l$ then $LB^* = (Z^h)^l$

end-if

Step 5: $Gap = (UB^l - LB^*)/UB^l \cdot 100\%$

If $Gap \le stoppingcriterion2$ then finish;

otherwise, go to Step 6

Step 6: Update Lagrangian multipliers according to (38)–(40);

$h = h + 1$, go to Step 1

---

6. Experiments and results

In this section, we present several experiments to test the performance of the Lagrangian relaxation solution algorithm and the performance of the AT system.

Fig. 5. Road network of the case study.

6.1. Application set-up

The model is applied to the case-study city of Delft, which is located in The Netherlands (Correia and van Arem, 2016a). The municipality has a total area of 24 km^2 and a population of about 100,000. Fig. 5 shows the simplified road network of Delft based on Google Map. In order to apply the model for studying the AT system in the city of Delft, the following data is needed: (1) the information of the requests; (2) the price rate of ATs, the cost of running the system and the penalties; (3) the road network information of Delft.

The mobility data were generated from the Dutch mobility dataset (MON 2007/2008) and transformed into the study of Correia and van Arem (2016a). It is provided by the Dutch government for research purposes. These travel data includes the origin and destination, travel mode, departure and arrival time for each sampled trip, which can be freely obtained online (Correia and van Arem, 2016b). For this application, 22,240 requests are considered represented by 1112 requests in the model, which means that an expansion coefficient of $\mu = 20$ is used (Correia and van Arem, 2016a). This is the result of filtering the database by selecting the trips done by cars and taxis who travelled inside the city of Delft during the surveyed day. At the same time, a fleet size of 500 ATs is tested in this application which are represented by 25 vehicles in the optimization model, using the same value of expansion coefficient.

Therefore one taxi satisfying one travel request represents 20 taxis satisfying 20 requests. The seating capacity of the ATs is set as 2, which indicates that an AT can at most take two shared ride passengers at the same time. The origins and destinations come from the mobility database as well as the departure times which are used as the desired departure time in the experiments. The maximum waiting time for each request is 22.5 min. The reason we set such a quite long maximum waiting time is that we want to provide a slack time window for the system to satisfy passenger requests. This will help to increase the percentage of satisfied requests. However, we are using a penalty so that the system reduces the waiting time. We consider a daytime operation of 15 h from 7:00–22:00 for the AT service. The time step of the optimization is 2.5 min. Each horizon contains 12 time steps ($T^h = 12$) and the rolling length is 6 time steps ($T^r = 6$). In practice, this means that when the system makes routing decisions for the ATs, it looks forward as far as 30 min and updates the ATs' status every 15 min.

The costs considered are as follows:

- $c_r = 1$ euro/min, referred to the price rate of Uber in Amsterdam and Rotterdam in The Netherlands ("Uber," 2017). We calculate the fare based on the shortest travel time, which means only the travel time dimension of price rate is sufficient.
- $c_f = 0.1$ euro/km, referred to Correia and van Arem (2016a).
- $c_v = 20$ euro/vehicle/day, referred to Liang et al. (2016).
- $c_p = 1$ euro/request.
- $c_d = 0.2$ euro/min.

The network has 66 road links and 46 nodes (see Fig. 5). Some of the links have one lane per direction and the other two lanes. The capacities were considered as 1500 and 3000 vehicles per hour per direction for one lane and two lanes roads respectively. The maximum speed was assumed to be 50 km/h and 70 km/h respectively for the lower and higher capacity links. In Fig. 5, the higher capacity links are shown in double lines and the lower ones are in single lines. The minimum travel time δ_{ij}^{min} on the link (i, j) is obtained from the free-flow speed. The maximum travel time δ_{ij}^{max} is computed based on a speed of 5 km/h. The initial locations of the vehicles are set to be at five nodes by constraint (11): node 27, node15, node 16, node 19, node 24, where they all have 100 vehicles available to balance the geographical distribution. The shortest travel times for the requests are computed by the shortest path method with the minimum link travel time. We set$K = 10$ meaning that for each running process of [$\boldsymbol{SP}2$] $10 + 1$ iterations are allowed, working with the stopping criterion that the error for each link travel time should be no more than 10%.

6.2. Base scenario

We programmed the model in Mosel language and solved the problem with the FICO® Xpress Solver in an i5 processor @ 3.10 GHz, 12.00 GB RAM computer under Windows 7 64-bit operating system. FICO® Xpress Solver provides optimization algorithms and technologies to solve linear, mixed integer and non-linear problems. The scenario with 500 ATs and a seating capacity of 2 is set as the base scenario.

The model [$\boldsymbol{OP^h}$] is firstly run for one horizon ($h = 48$) as an example to illustrate how it works the type of results that can be obtained. The 48th horizon is chosen because there are 400 travel requests which is close to the average number of requests per horizon. We run the model from the first horizon to the 47th horizon and use the vehicle status and demand satisfaction information from the 47th horizon to guarantee the continuity. Then model [$\boldsymbol{OP^h}$] is solved for the 48th horizon with 300 Lagrangian iterations.

The graphical and numerical results are shown in Fig. 6 and Table 4. The upper bound of [$\boldsymbol{OP^h}$] drops rapidly within the first 50 iterations from about 1000 to 328.4, while the lower bound also has an obvious growth at the same time from about 10 to 209.4. The gap between the current best solution and the current best bound is about 36%. From the 51st iteration to the 100th, the model begins to converge at a stable speed and the upper bound and the lower bound get closer, which narrows the gap to 25%. After iteration 100, the convergence slows down but the upper bound and the lower bound keep improving. The lower bounds from the 100th iteration to the 150th iteration are all the same meaning that during this time the model cannot find a better feasible solution

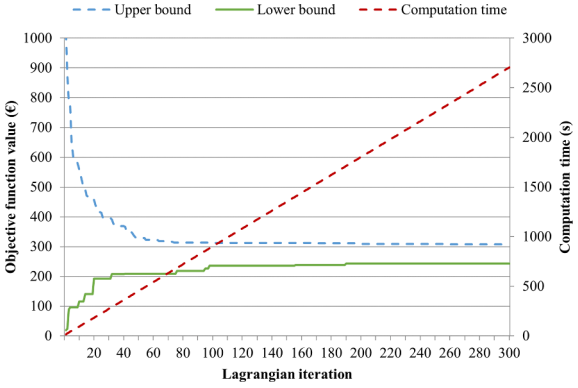


Fig. 6. Computation results for one horizon.

Table 4
Computation results for one horizon.

Until iteration	10	20	30	40	50	60	70	80	90	100
Upper bound	558.1	457.3	396.2	368.3	328.4	321.4	316.8	313.8	313.8	313.8
Lower bound	115.4	192.9	192.9	207.4	209.4	209.4	209.4	218.4	218.4	235.4
Gap	79%	58%	51%	44%	36%	35%	34%	30%	30%	25%
Until iteration	**110**	**120**	**130**	**140**	**150**	**160**	**170**	**180**	**190**	**200**
Upper bound	312.9	312.3	312.3	312.3	312.3	312.2	311.7	311.7	311.5	308.9
Lower bound	235.4	235.4	235.4	235.4	235.4	238.4	238.4	238.4	242.4	242.4
Gap	25%	25%	25%	25%	25%	24%	24%	24%	22%	22%
Until iteration	**210**	**220**	**230**	**240**	**250**	**260**	**270**	**280**	**290**	**300**
Upper bound	308.9	308.8	308.8	308.6	308.5	308.2	308.2	308.0	308.0	308.0
Lower bound	242.4	242.4	242.4	242.4	242.4	242.4	242.4	242.4	242.4	242.4
Gap	22%	21%	21%	21%	21%	21%	21%	21%	21%	21%

to the $[\boldsymbol{OP^h}]$ to replace the current one. However, the upper bound keeps going down and this contributes to closing the gap. After 190th iteration, the lower bound does not change until this solving process reaches 300 iterations, but the upper bound slightly increases, with the final gap being 21%.

The computation time is proportional to the number of Lagrangian iterations. With more iterations, it has a higher probability to find a better feasible solution and a lower upper bound, which is able to show that this solution is closer to the optimal one. However, more iterations also bring longer computation time. In this one horizon example, iteration 50, 100, 200 and 300 are reasonable breakpoints: the gaps are 36%, 25%, 22%, and 21% respectively, showing that the quality of the solutions is acceptable. Therefore, we decide to choose $L = 50$, 100, 200 and 300 as the maximum number of Lagrangian iterations to run all the horizons and compare the final results of the base scenario. The computation results with different Lagrangian iterations are presented in Table 5.

When 50 Lagrangian iterations are used for each horizon, the model spends 15.9 min per horizon to obtain a feasible solution with the average gap of 32.4% and the median value of the horizon gap 29.5%. While if we increase the iteration number to 100, the model spends 34.3 min per horizon to find an acceptable feasible solution. At the same time, the average gap per horizon is decreased from 32.4% to 26.8%, and the median of the gaps is also decreased from 29.5% to 23.5%. This demonstrates that increasing the number of Lagrangian iterations helps the model to close the gap, while along with the computation time growing. When 200 Lagrangian iterations are applied, the average gap keeps decrease to 23.0%, while the median of the horizon gaps has a more significant drop from 23.5% to 9.0%. This means that half of the horizons have the optimization gap below or equal to 9.0% and among the rest half horizons, some of them are difficult to solve which makes the average gap higher than the median. When we increase Lagrangian iterations from 200 to 300, the average gap has a slight decrease from 23.0% to 22.5% while the computation time has a significant increase from 57.6 min to 97.6 min per horizon. Moreover, the median of the gaps for all the horizons increases from 9.0% to 15.0%. This is because when the results from the previous horizon are different, then the initial status of such AT system will change in the next horizon and that will turn the problem into a different one in the next horizon.

In the following experiments, we believe that iteration 200 is an acceptable stopping criterion to obtain a feasible solution with good quality. Additionally, the iterative process will stop when the gap is lower or equal to 1% even if the number of iterations is lower than 200, which is another stopping criterion. Even though the computation time of this model is not fast enough to handle a real-time problem, we still can expect that the advances of computer technologies and new algorithms can accelerate significantly the solving of this model in the future.

The time distribution of the satisfied requests using 200 Lagrangian iterations is shown in Fig. 7. The demand is not uniformly distributed during the day, which creates the “peak hours” and the “off-peak hours” in the rolling horizon framework. In this scenario, 13,460 passengers in total are served out of 22240, meaning that 60.5% of the requests are served by ATs. In the morning peak hours, the AT system has a relatively low satisfaction rate. The afternoon peak hours have a similar tendency. However, the system provides a higher share of ATs for the low-demand time periods and some of them have a 100% satisfaction rate. This figure illustrates that the high demand for the morning and afternoon peak hours exceeds the transport capacity of the system. It also means that in the off-peak horizons, 500 ATs are redundant and generate idle time. Moreover, the objective function of $\boldsymbol{[OP^h]}$ is to maximize

Table 5
Results for the base scenario with 50, 100, 200 and 300 Lagrangian iterations.

Number of iterations	Average computation time per horizon (min)	Average gap per horizon	Median of the horizon gaps
50	15.9	32.4%	29.5%
100	34.3	26.8%	23.5%
200	57.6	23.0%	9.0%
300	97.6	22.5%	15.0%

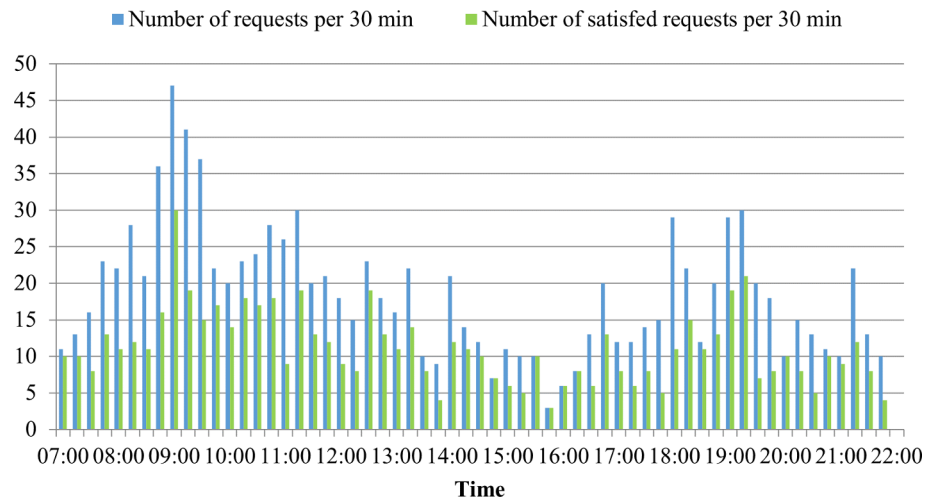


Fig. 7. Requests distribution.

the daily profit of the AT company, which means that the system will select the requests to be served when they bring more profit. The objective function also considers the service quality by adding the penalty for rejected requests and the delivery delay. Therefore, the AT system aims at having a higher AT share to gain more revenue and keep a higher service quality to avoid more penalties. Sections 6.3 and 6.4 presents more detailed discussions on this.

6.3. Optimization results

To test the performance of the proposed model, we vary the scenarios in two dimensions: the number of ATs and the seating capacity of each AT. We also vary the values of AT price rate, rejection penalty and delay penalty to conduct a sensitivity analysis with respect to the cost components. The descriptions for all the scenarios are shown in Table 6 and the optimization results can be seen in Table 7. AT share is defined as the number of requests ATs are serving divided by the total number of requests. "Percentage of time in service" is calculated by dividing the total travel time when the ATs are transporting passengers by the total time for the whole operation period. The delay is computed as the time spent by each traveller from the desired departure time to the real arrival time minus the shortest travel time. It includes the departure delay (waiting time) and the congestion delay (real travel time minus the shortest travel time).

6.3.1. Fleet size variation

Scenarios 0, 1 and 2 are tested with different fleet sizes: 400, 500 and 600 (Table 7). When there are only 400 ATs in the city in scenario 1, the system produces 84.2 $\times$ 10^3 euros as daily profit, which is 6% lower than the base scenario with 500 ATs, while with 600 ATs the system has 1% higher daily profit. This is because of the different components considered in the system's profit. When more ATs are available for the passengers, they will serve more passengers' requests and the AT company will get more revenue from them, which can be seen from the columns of "Total revenue" and "Number of satisfied requests". At the same time, the penalty paid for rejecting requests will be lower, which also contributes to the system profitability. However, the AT's depreciation cost and the fuel cost also increase with having more vehicles in the system, since these costs are almost proportional to the fleet size.

In these scenarios, the fleet size is a critical parameter with respect to the AT share in urban mobility. When there are only 400

Table 6
Scenario description.

Scenario	Description	V	$vcap$	c_r euro/min	c_p euro/request	c_d euro/min
0	**Base**	**500**	**2**	**1**	**1**	**0.2**
1	Fewer ATs in the system	400	2	1	1	0.2
2	More ATs in the system	600	2	1	1	0.2
3	Lower seating capacity (no ride-sharing)	500	1	1	1	0.2
4	Higher seating capacity for ride-sharing	500	3	1	1	0.2
5	Lower price rate	500	2	0.5	1	0.2
6	Higher price rate	500	2	1.5	1	0.2
7	No penalty for request rejection	500	2	1	0	0.2
8	Higher penalty for request rejection	500	2	1	2	0.2
9	No penalty for delivery delay	500	2	1	1	0.0
10	Higher penalty for delivery delay	500	2	1	1	0.4

Table 7
Optimization results for the reference scenarios.

Scenario	Profit per day ($\times 10^3$ euros)	Total revenue ($\times 10^3$ euros)	Total fuel cost ($\times 10^3$ euros)	Total penalty cost for rejected requests ($\times 10^3$ euros)	Total depreciation cost ($\times 10^3$ euros)	Total penalty cost for delay ($\times 10^3$ euros)	Number of satisfied requests	AT share	Avg. satisfied requests Per AT	Avg. travel time in service per AT (min)	Percentage of time in service	Avg. price per satisfied request (euros)	Avg.travel time per satisfied request (min)	Avg. delay per satisfied request (min)	Avg. waiting time per satisfied request (min)	Avg. congestion delay per satisfied request (min)
0) Base	**89.6**	**192.1**	**45.1**	**8.8**	**10.0**	**38.6**	**13460**	**60.5%**	**26.9**	**415.1**	**46.1%**	**14.3**	**15.4**	**14.3**	**13.0**	**1.3**
1) Fewer ATs	84.2	174.8	36.0	10.0	8.0	36.6	12220	54.9%	30.6	471.5	52.4%	14.3	15.4	15.0	13.7	1.3
2) More ATs	90.4	205.1	54.0	7.9	12.0	40.8	14360	64.6%	23.9	371.3	41.3%	14.3	15.5	14.2	12.8	1.4
3) Lower seating capacity	43.7	140.3	45.1	12.5	10.0	29.0	9700	43.6%	19.4	299.0	33.2%	14.5	15.4	14.9	14.0	0.9
4) Higher seating capacity	118.4	224.0	44.8	6.7	10.0	44.1	15560	70.0%	31.1	489.1	54.3%	14.4	15.7	14.2	12.9	1.3
5) Lower price rate	-7.0	91.0	45.3	9.7	10.0	33.0	12560	56.5%	25.1	388.6	43.2%	7.2	15.5	13.1	12.1	1.0
6) Higher price rate	151.1	255.4	44.8	8.3	10.0	41.2	13980	62.9%	28.0	430.8	47.9%	18.3	15.4	14.7	13.4	1.3
7) No penalty for rejection	91.4	181.2	45.1	0.0	10.0	34.7	12480	56.1%	25.0	394.8	43.9%	14.5	15.8	13.9	12.6	1.3
8) Higher pen. for rejection	83.7	195.5	45.0	17.2	10.0	39.6	13640	61.3%	27.3	419.0	46.6%	14.3	15.4	14.5	13.2	1.3
9) No pen. for delay	139.2	202.5	45.3	8.0	10.0	0.0	14280	64.2%	28.6	519.3	57.7%	14.2	18.2	20.4	16.4	4.0
10) Higher pen. for delay	28.5	146.3	44.7	11.9	10.0	51.2	10300	46.3%	20.6	310.5	34.5%	14.2	15.1	12.4	11.6	0.8

vehicles in the taxi system, 12,220 requests are satisfied in total. In this case, the AT share, which can also be considered as the satisfied rate, is the lowest among all the scenarios. When the fleet size increases to 500, the AT share increases by 5.6%. This trend continues in the next scenario with 600 ATs, but the growth rate is falling. This means that with the second 100 vehicles added the system cannot provide a significant growth on service coverage, which also means that from that moment on, other indicators become critical in respect to the improvement of daily profit, i.e. the service quality.

With respect to vehicle usage, the most efficient scenario is the one with 400 ATs, with 30.6 requests per vehicle. This can also be seen in the column “Average travel time in service”. The percentage of time in service is 52.4%, which is the highest of the three. The 600 ATs scenario has the lowest percentage of time in service (41.3%), which means that 58.7% of the total operation time ATs are idle. The added ATs are redundant in off-peak horizons, while in the peak horizons they are more helpful and can contribute to a higher AT share. On average, each passenger spends 14.3 euros per trip and stays 15.4 min inside the vehicle in the 500 vehicle scenario. These values almost the same for scenario 1 and 2, meaning that the fleet size does not have a significant influence on these two indicators.

The average delay per satisfied request keeps decreasing from the fewer-ATs scenario to the more-ATs one. This happens because of the different average waiting time for each scenario and the average congestion time. The variation of the fleet size has a significant influence on the waiting time of the satisfied requests for the scenarios of 400 and 500 ATs, with 0.7 min reduction. While from 500 ATs to 600 ATs, it becomes only 0.1 min. This demonstrates that the first additional 100 ATs helps a lot regarding the waiting time and accomplish more requests to increase service performance. In this case, the AT’s service quality is improved from the passengers’ perspective, which is consistent with the setting of the objective function (29). However, the second additional 100 ATs does not contribute a lot to the reduction in the waiting time for departure, but they indeed satisfy more requests which increase the AT share to 64.6%.

6.3.2. Ride-sharing variation

In these experiments, we apply the flat AT price rate for the different seating capacities, in order to analyse the interaction between the level of ride-sharing and the system profitability, the AT share and the service quality.

The seating capacity indicates the maximum number of passengers allowed to share a ride at the same time. When it varies from 1 to 3, it also increases the total transport capacity of the AT system. It is obvious from Table 7 that the improvement of the system profitability is significant when the system changes its serving scheme from individual AT service to ride-sharing AT service. When the seating capacity increased from 2 to 3, the improvement of the total profit happens but not with the same magnitude as when it goes from 1 to 2. The AT service company earns 45.9×10^3 euros more in allowing two passengers to share a ride, compared with no ride-sharing. Looking at the constituent parts of the profit, the increase in the total revenue is the main contributor to the profit. This is because when the seating capacity of the ATs is doubled, the system is able to serve more travel requests, which can be seen from the column “Number of satisfied requests” and “AT share”. On one hand, it will bring more revenue to the AT company; while on the other hand, it reduces the penalty paid to the rejected passengers who have to use public transport as an alternative to accomplish their trips. However, when three passengers are allowed to share a ride the AT system serves 2100 more requests which provide 31.9×10^3 euros more to the system revenue, 2.1×10^3 euros less on the rejection penalty cost and 5.5×10^3 euros less on the delay penalty. This improvement is not as considerable as the one obtained going from no ride-sharing to two seat sharing. This means that the system is not able to serve more requests, since other reasons become a constraint for the demand satisfaction, e.g. the departure and arrival time windows.

At the same time, the usage of the vehicles is also improved. The two-seat scenario has 7.5 more requests served per AT and 116.1 min more in service than the one-seat scenario. With the doubled value of seating capacity, these values are not as doubled as is

could be expected. This happens because there are still some other constraints for satisfying the requests, e.g. hard constraints like the request's individual time window and soft constraints like the delay penalty due to the ride-sharing. This also means that the ATs are not always taking shared-ride passengers. The trend also happens in the scenario with 3 ride-sharing passengers, but the increases are not as significant as the previous one.

Due to the sufficient transporting capacity, the average delay drops from 14.9 min to 14.3 min and then to 14.2 min, which indicates the improvement of the service level offered to clients. To be more specific, the waiting time keeps going down among the three scenarios; while the congestion delay has an exceptional case: 1.3 min for scenarios 0 and 4, which is 0.4 min higher than scenario 3. Even though, the AT service quality regarding the delay is improved because of the ride-sharing.

6.3.3. Sensitivity analysis on price rate

The price rate has a critical impact on the AT system's daily profit. It is easy to see that the number of satisfied requests by the ATs increases from 12,560 to 13,460 and then to 13,980 when increasing the price rate from 0.5 euro/min to 1 euro/min then to 1.5 euro/min. The same trend also happens with the "Total penalty cost for rejected requests" and "Total penalty cost for delay", showing that the monetary benefits gaining from the increase of the AT share remain at a relatively low level. The reason for this is that when the system can have more revenue from the same requests by increasing the price rate, the difference between the revenue and the cost for each satisfied request is more obvious. Therefore, the system has a greater motivation to satisfy more requests. At the same time, more requests may lead to a more congested road network, which increases the penalty paid for the travel delay. So the system should find a balance between the profit benefit from the higher AT share and the profit loss from the delivery delay.

Since the fleet size is a constant for these three scenarios, the fuel cost and the vehicle depreciation costs have almost the same values in the objective function. Therefore, the difference between daily profits results primarily from the revenue earned from AT passengers. When the price is 0.5 euros/min, the AT company can obtain 91.0×10^3 euros as revenue; while this value is almost doubled to 192.1×10^3 euros when the price increases to 1 euro/min. Similarly, the revenue of the 1.5 euros/min scenario is almost three times the one obtained with 0.5 euros/min. This means that the system revenue is approximately proportional to the price rate of the AT service. This can also be seen in the column "Avg. price per satisfied request", where the average monetary cost for one request is doubled from scenario 5 to scenario 0. While in scenario 6 the average price per request is lower than three times of that price in scenario 5, indicating that when the price is higher, the system tends to serve more short (cheap) requests. However, the revenue for scenario 5 is not high enough to cover the other costs for the system, leading to a negative profit as the final result.

6.3.4. Sensitivity analysis on rejection penalty

The rejection penalty is charged when a potential request is rejected by the AT system and the value of it is set according to the cost of a public transport alternative. This penalty is aimed at improving service performance and encourage the system to take more passengers. It also allows the AT system to make the decision that when some requests are either unprofitable or un-satisfiable, it can reject these requests.

The rejection penalty brings profit loss to the system. The profit decreases 1.8×10^3 euros when paying 1 euro for each rejected request when compared to not having a penalty. This is due to the rejection penalty of 8.8×10^3 euros, the additional 3.9×10^3 euros for the delay penalty and the extra revenue of 10.9×10^3 euros. If the rejection penalty increases to 2 euros per unsatisfied request, it brings 3.4×10^3 euros more to the revenue, 8.4×10^3 euros more to the rejection penalty cost and 1.0×10^3 euros more to the delay cost, which in total contributes 5.9×10^3 euros less to the daily profit of the AT system.

The rejection penalty plays a role in promoting the growth of the AT share, increasing from 12,480 to 13,460 requests, and from 13,460 to 13,640 requests. The revenues of these served requests have slight growth for these three scenarios. However, the average price per satisfied request of the scenario with no penalty for rejection is the highest, indicating that the average shortest travel times for the satisfied requests in scenarios 0 and 8 are shorter than scenario 7. This happens because the system tends to satisfy requests that have shorter travel times to avoid lower rejection costs. But the system profit still cannot benefit from this penalty policy, since it is not helpful in increasing the revenue from the passengers.

The travel delay firstly increases from 13.9 min to 14.3 min, then decreases to 14.5 min, for scenarios 7, 0 and 8. This trend also happens to the "Average waiting time per satisfied request" and the "Average congestion delay per satisfied request". The first increase from no penalty to one euro is due to the growth in the total delay penalty and the average satisfied requests per AT because the ATs are busier to satisfy more requests and this leads to more delay for each request. A small increase happens as a result of increasing one euro to two euros the rejection penalty, indicating that it is better to serve more requests but with a good service quality to avoid the costs for the added delay. In this way, the daily profit of the system is maintained at a specific level.

6.3.5. Sensitivity analysis on the delay penalty

The daily profit is sensitive to the delay penalty. When there is no charge for late arrival, the system only needs to guarantee each satisfied request to be picked up and dropped off within its allowed departure and arrival time window. However, if the AT company needs to pay for the time delay of delivering the passenger, the system not only needs to comply with the time windows but it also needs to provide arrivals that happen as early as possible. This penalty reflects the service quality of the AT system.

For the scenario with no delay penalty, the system has the highest daily profit (139.2×10^3 euros), the highest AT share (64.2%) and the highest average delay (4.0 min) in transporting passengers among scenario 9, 0 and 10. This is the most profitable scenario among the three due to the high revenues from the satisfied requests and the zero penalty cost for the time delay. However, the other two scenarios have 820 and 3980 fewer requests being satisfied by ATs, which generate lower revenue for the system. This also makes the system pay the lowest value for the rejection penalty when there is no delay penalty. Regarding the

vehicle usage, the system becomes less efficient when increasing the value of delay penalty, which can be seen in the "Average satisfied requests per AT" and the "Percentage of time in service". This is not surprising since the same number of ATs are used to provide a different number of satisfied requests. The time delay has a sharp decrease (from 20.4 min to 12.4 min) with the increase of the delay penalty. This tendency also occurs in the waiting time and the congestion delay. With the high penalty for the time delay, the system profitability is quite sensitive to the late arrival. In some extreme cases, if a request is delivered with a long delay time, it may happen that the delay penalty is higher than the revenue obtained from this request, which means satisfying this passenger will be damaging for the profit. Under these circumstances, the system decides to serve fewer requests guaranteeing the service level (short delay time) to protect the profit. Therefore, the delay penalty is an appropriate control parameter to assure the service quality being provided by the ATs.

7. Conclusion

AVs have drawn great attention in recent decades evolving from just a concept to reality with several pilot studies. An emphasis has been put on technology in recent research focusing on transport reliability and safety, and the interactions between the AVs and the other vehicles, pedestrians and cyclists. Nevertheless, some questions related to the application of AVs have hardly been addressed, such as their use as public transport.

This paper proposed an INLP model $[OP^h]$ to study the DARP of ATs under dynamic travel times created by the ATs themselves. The model has as its main objective to maximize the total daily profit of such system by deciding on each AT's routing with real-time information. The penalties of the rejected requests and the delivery delay are considered in the objective function to guarantee the performance of the AT service. We argue that this type of model is needed in future situations in which a large number of requests and a large number of vehicles are involved. Therefore, the travel time of the ATs is modelled as a function of the traffic flow of the ATs themselves on the road links in the constraints. The model considers the demand as real-time requests which are managed via a rolling horizon structure thus allowing the vehicles' routing being optimized horizon by horizon. Even though this framework makes it possible to decrease the scale of the problem in each optimization process, the proposed model still involves a large number of integer variables, which makes it an NP-hard one. We develop a customized Lagrangian relaxation algorithm to solve the mathematical model, which decomposes the $[OP^h]$ into two sub-problems. Based on the solutions gained from the two sub-problems, we are able to find the best bound of the original problem and the current best solution, which is feasible to the original problem. In this way, it is not only possible to find a near-optimal solution but also to have a notion of how good this solution is.

The model and the solving algorithm was applied to the case study city of Delft, The Netherlands, with 15 h service period and 22,240 travel requests generating from the road network with 46 nodes and 66 links. From that application, it was possible to make several conclusions.

The customized Lagrangian solving algorithm is able to solve the proposed NP-hard problem and obtain a good quality feasible solution within the acceptable computation time. The solving times are still high enough to forbid their application in real-time systems but we believe that this is an initial version of the type of algorithms that must be built and run in the future. The fleet size is an important factor in system profitability and the satisfaction of the requests. When the fleet increases, the number of satisfied requests is not increasing with the same rate, which demonstrates that the fleet size is not the only reason to constrain the satisfaction of the requests. In fact, the uneven time distribution of the demand and the strict departure and arrival time windows are all important factors in influencing the AT share. The model shows that the AT passengers will experience a delivery delay when considering the traffic congestion generated by the AT themselves. With ride-sharing, the AT system has more capacity to provide better service regarding the number of satisfied requests and the average delay time. The system profitability is sensitive to the price rate of the AT service. The different values of the rejection penalty will lead to changes in the ATs' daily profit and the number of satisfied requests. The delay penalty seems to be a proper control parameter to guarantee a certain service quality being offered by the ATs systems. When this penalty is not considered, the AT system will have a higher AT share, along with a longer delivery delay for the satisfied requests.

The current version of the optimization model is not practice-ready, due to the computation time and the computation gap between the upper and lower bounds. However, what we are providing is a conceptual model to address ATs' DARP considering traffic congestion through an exact mathematical optimization program. This formulation can be the basis for future work. It can be used to develop heuristics for the ATs' DARP problem, which may accelerate the optimization process. In addition, more efforts can be devoted to improving the solution algorithm and close the computation gap of the proposed model. In this research, the AT system covers 50%-70% of the total transport demand, with the objective to maximize the AT company's profit. Future research could select the share of ATs as an objective and investigate the changes in performance. There is also the possibility of involving choice modelling to analyse people's preferences for using ATs. This would allow running a sensitivity analysis on the influence of the AT price rate on the potential demand and the overall system profitability.

CRediT authorship contribution statement

Xiao Liang: Conceptualization, Methodology, Software, Validation, Formal analysis, Data curation, Writing - original draft, Visualization. **Gonçalo Homem de Almeida Correia:** Conceptualization, Methodology, Software, Resources, Data curation, Writing - review & editing. **Kun An:** Conceptualization, Methodology, Writing - review & editing. **Bart van Arem:** Methodology, Resources, Writing - review & editing, Supervision.

Acknowledgements

This work has been supported by ProRail, the train infrastructure manager of The Netherlands. The first author has been sponsored by the China Scholarship Council (CSC).

Appendix A. Supplementary material

Supplementary data to this article can be found online at https://doi.org/10.1016/j.trc.2020.01.024.

References

Agatz, N., Erera, A., Savelsbergh, M., Wang, X., 2012. Optimization for dynamic ride-sharing: a review. Eur. J. Oper. Res. 223, 295–303. https://doi.org/10.1016/j.ejor.2012.05.028.

Agatz, N.A.H., Erera, A.L., Savelsbergh, M.W.P., Wang, X., 2011. Dynamic ride-sharing: A simulation study in metro Atlanta. Transp. Res. Part B Methodol. 45, 1450–1464. https://doi.org/10.1016/j.trb.2011.05.017.

An, K., Xie, S., Ouyang, Y., 2017. Reliable sensor location for object positioning and surveillance via trilateration. Transp. Res. Part B Methodol. 23, 228–245. https://doi.org/10.1016/j.trb.2017.11.012.

Angelopoulos, A., Gavalas, D., Konstantopoulos, C., Kypriadis, D., Pantziou, G., 2018. Incentivized vehicle relocation in vehicle sharing systems. Transp. Res. Part C Emerg. Technol. 97, 175–193. https://doi.org/10.1016/j.trc.2018.10.016.

Bai, Y., Hwang, T., Kang, S., Ouyang, Y., 2011. Biofuel refinery location and supply chain planning under traffic congestion. Transp. Res. Part B Methodol. 45, 162–175. https://doi.org/10.1016/j.trb.2010.04.006.

Balac, M., Becker, H., Ciari, F., Axhausen, K.W., 2019. Modeling competing free-floating carsharing operators – A case study for Zurich, Switzerland. Transp. Res. Part C Emerg. Technol. 98, 101–117. https://doi.org/10.1016/j.trc.2018.11.011.

Beckmann, M.J., McGuire, C.B., Winsten, C.B., 1955. Studies in the Economics of Transportation. Yale Univ. Press, New Haven.

Celsor, C., Millard-Ball, A., 2007. Where does carsharing work?: using geographic information systems to assess market potential. Transp. Res. Rec. J. Transp. Res. Board 1992, 61–69. https://doi.org/10.3141/1992-08.

Cordeau, J.F., Laporte, G., 2007. The dial-a-ride problem: models and algorithms. Ann. Oper. Res. 153, 29–46. https://doi.org/10.1007/s10479-007-0170-8.

Correia, G., Viegas, J.M., 2011. Carpooling and carpool clubs: Clarifying concepts and assessing value enhancement possibilities through a Stated Preference web survey in Lisbon, Portugal. Transp. Res. Part A Policy Pract. 45, 81–90. https://doi.org/10.1016/j.tra.2010.11.001.

Correia, G.H.de A., van Arem, B., 2016a. Solving the User Optimum Privately Owned Automated Vehicles Assignment Problem (UO-POAVAP): A model to explore the impacts of self-driving vehicles on urban mobility. Transp. Res. Part B Methodol. 87, 64–88. https://doi.org/10.1016/j.trb.2016.03.002.

Correia, G.H.de A., van Arem, B., 2016b. Trips and network of the case-study city of Delft. https://doi.org/10.13140/RG.2.2.11097.83043.

Dafermos, S., Sparrow, F., 1969. Traffic assignment problem for a general network. U S Bur Stand. Res. Sci.

Fagnant, D.J., Kockelman, K.M., 2018. Dynamic ride-sharing and fleet sizing for a system of shared autonomous vehicles in Austin, Texas. Transport. (Amst.) 45, 143–158. https://doi.org/10.1007/s11116-016-9729-z.

Fagnant, D.J., Kockelman, K.M., 2014. The travel and environmental implications of shared autonomous vehicles, using agent-based model scenarios. Transp. Res. Part C Emerg. Technol. 40, 1–13. https://doi.org/10.1016/j.trc.2013.12.001.

Fisher, M.L., 2004. The lagrangian relaxation method for solving integer programming problems. Manage. Sci. 50, 1861–1871. https://doi.org/10.1287/mnsc.1040.0263.

Fisher, M.L., 1981. The lagrangian relaxation method for solving integer programming problems. Manage. Sci. 27, 1–18. https://doi.org/10.1287/mnsc.27.1.1.

Furuhata, M., Dessouky, M., Ordóñez, F., Brunet, M.E., Wang, X., Koenig, S., 2013. Ridesharing: The state-of-the-art and future directions. Transp. Res. Part B Methodol. 57, 28–46. https://doi.org/10.1016/j.trb.2013.08.012.

Ho, S.C., Szeto, W.Y., Kuo, Y.H., Leung, J.M.Y., Petering, M., Tou, T.W.H., 2018. A survey of dial-a-ride problems: literature review and recent developments. Transp. Res. Part B Methodol. 111, 395–421. https://doi.org/10.1016/j.trb.2018.02.001.

Hoogendoorn, R., van Arerm, B., Hoogendoorn, S., 2014. Automated driving, traffic flow efficiency, and human factors. Transp. Res. Rec. J. Transp. Res. Board 2422, 113–120. https://doi.org/10.3141/2422-13.

Huang, K., Correia, G.H.de A., An, K., 2018. Solving the station-based one-way carsharing network planning problem with relocations and non-linear demand. Transp. Res. Part C Emerg. Technol. 90, 1–17. https://doi.org/10.1016/j.trc.2018.02.020.

Hyland, M., Mahmassani, H., 2018. Dynamic autonomous vehicle fleet operations: optimization-based strategies to assign AVs to immediate traveler demand requests. Transp. Res. Part C Emerg. Technol. 92, 278–297. https://doi.org/10.1016/j.trc.2018.05.003.

Imai, A., Nishimura, E., Current, J., 2007. A Lagrangian relaxation-based heuristic for the vehicle routing with full container load. Eur. J. Oper. Res. 176, 87–105. https://doi.org/10.1016/j.ejor.2005.06.044.

Jorge, D., Barnhart, C., de Almeida Correia, G.H., 2015. Assessing the viability of enabling a round-trip carsharing system to accept one-way trips: application to Logan Airport in Boston. Transp. Res. Part C Emerg. Technol. 56, 359–372. https://doi.org/10.1016/j.trc.2015.04.020.

Kaufman, D.E., Nonis, J., Smith, R.L., 1992. A mixed integer linear programming formulation of the dynamic traffic assignment problem. In: Syst. Man Cybern. 1992., IEEE Int. Conf. pp. 232–235. https://doi.org/10.1109/ICSMC.1992.271771.

KPMG, 2012. Self-driving cars: the next revolution.

Krueger, R., Rashidi, T.H., Rose, J.M., 2016. Preferences for shared autonomous vehicles. Transp. Res. Part C Emerg. Technol. 69, 343–355. https://doi.org/10.1016/j.trc.2016.06.015.

Laporte, G., 2009. Fifty years of vehicle routing. Transp. Sci. 43, 408–416. https://doi.org/10.1287/trsc.1090.0301.

Lei, C., Ouyang, Y., 2018. Continuous approximation for demand balancing in solving large-scale one-commodity pickup and delivery problems. Transp. Res. Part B 109, 90–109. https://doi.org/10.1016/j.trb.2018.01.009.

Levin, M.W., 2017. Congestion-aware system optimal route choice for shared autonomous vehicles. Transp. Res. Part C Emerg. Technol. 82, 229–247. https://doi.org/10.1016/j.trc.2017.06.020.

Li, B., Krushinsky, D., Van Woensel, T., Reijers, H.A., 2016. The Share-a-Ride problem with stochastic travel times and stochastic delivery locations. Transp. Res. Part C Emerg. Technol. 67, 95–108. https://doi.org/10.1016/j.trc.2016.01.014.

Liang, X., Correia, G.H.de A., van Arem, B., 2018. Applying a model for trip assignment and dynamic routing of automated taxis with congestion: system performance in the city of Delft, The Netherlands. 036119811875804. Transp. Res. Rec. J Transp. Res. Board. https://doi.org/10.1177/0361198118758048.

Liang, X., Correia, G.H.de A., van Arem, B., 2016. Optimizing the service area and trip selection of an electric automated taxi system used for the last mile of train trips. Transp. Res. Part E Logist. Transp. Rev. 93, 115–129. https://doi.org/10.1016/j.tre.2016.05.006.

Luo, Y., Schonfeld, P., 2011. Online Rejected-Reinsertion Heuristics for Dynamic Multivehicle Dial-a-Ride Problem. Transp. Res. Rec. J. Transp. Res. Board 2218, 59–67. https://doi.org/10.3141/2218-07.

Ma, J., Li, X., Zhou, F., Hao, W., 2017. Designing optimal autonomous vehicle sharing and reservation systems: A linear programming approach. Transp. Res. Part C Emerg. Technol. 84, 124–141. https://doi.org/10.1016/j.trc.2017.08.022.

Martinez, L.M., Viegas, J.M., 2017. Assessing the impacts of deploying a shared self-driving urban mobility system: An agent-based model applied to the city of Lisbon. Portugal. Int. J. Transp. Sci. Technol. 6, 1–15. https://doi.org/10.1016/j.ijtst.2017.05.005.

Nieuwenhuijsen, J., Correia, G.H.de A., Milakis, D., van Arem, B., van Daalen, E., 2018. Towards a quantitative method to analyze the long-term innovation diffusion of automated vehicles technology using system dynamics. Transp. Res. Part C Emerg. Technol. 86, 300–327. https://doi.org/10.1016/j.trc.2017.11.016.

Pillac, V., Gendreau, M., Guéret, C., Medaglia, A.L., 2013. A review of dynamic vehicle routing problems. Eur. J. Oper. Res. 225, 1–11. https://doi.org/10.1016/j.ejor.2012.08.015.

Psaraftis, H., 1980. A Dynamic Programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. Transp. Sci. 14, 130–154. https://doi.org/10.1287/trsc.14.2.130.

Psaraftis, H.N., 1988. Vehicle routing: Methods and studies. Dyn. Veh. Routing Probl. North Holland, Amsterdam, Netherlands, pp. 223–248.

SAE International, 2014. Summary of SAE International's level of Driving Automationn for On-Road Vehicles.

Schaller, B., 2018. The New Automobility: Lyft, Uber and the Future of American Cities.

Schuster, T., Byrne, J., Corbett, J., Schreuder, Y., 2005. Assessing the Potential Extent of Carsharing: A New Method and Its Implications. Transp. Res. Rec. J. Transp. Res. Board 1927, 174–181. https://doi.org/10.3141/1927-20.

Shaheen, S., Sperling, D., Wagner, C., 1999. A Short History of Carsharing in the 90's. J. World Transp. Policy Pract. 5, 16–37. https://doi.org/10.1007/s11116-007-9132-x.

Shen, Q., Chu, F., Chen, H., 2011. A Lagrangian relaxation approach for a multi-mode inventory routing problem with transshipment in crude oil transportation. Comput. Chem. Eng. 35, 2113–2123. https://doi.org/10.1016/j.compchemeng.2011.01.005.

Spieser, K., Treleaven, K., Zhang, R., Frazzoli, E., Morton, D., Pavone, M., 2014. Toward a Systematic Approach to the Design and Evaluation of Automated Mobility-on-Demand Systems: A Case Study in Singapore. Road Vehicle Automation 229–245. https://doi.org/10.1007/978-3-319-05990-7_20.

Uber [WWW Document], 2017. URL https://www.uber.com/.

Wen, J., Chen, Y.X., Nassir, N., Zhao, J., 2018. Transit-oriented autonomous vehicle operation with integrated demand-supply interaction. Transp. Res. Part C Emerg. Technol. 97, 216–234. https://doi.org/10.1016/j.trc.2018.10.018.

Yang, J., Jaillet, P., Mahmassani, H., 1999. On-Line Algorithms for Truck Fleet Assignment and Scheduling Under Real-Time Information. Transp. Res. Rec. 1667, 107–113. https://doi.org/10.3141/1667-13.

Yap, M.D., Correia, G., van Arem, B., 2016. Preferences of travellers for using automated vehicles as last mile public transport of multimodal train trips. Transp. Res. Part A Policy Pract. 94, 1–16. https://doi.org/10.1016/j.tra.2016.09.003.