MILP models for the Dial-a-Ride problem with transfers

Pierotti, Jacopo; van Essen, J. Theresia

DOI
10.1016/j.ejtl.2021.100037

Publication date
2021

Document Version
Final published version

Published in
EURO Journal on Transportation and Logistics

EURO Journal on Transportation and Logistics 10 (2021) 100037

Contents lists available at ScienceDirect

EURO Journal on Transportation and Logistics

journal homepage: www.journals.elsevier.com/euro-journal-on-transportation-and-logistics

MILP models for the Dial-a-ride problem with transfers

Jacopo Pierotti *, J. Theresia van Essen

Delft Institute of Applied Mathematics, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Mekelweg 4, 2628 CD, Delft, the Netherlands

ARTICLE INFO

Keywords:
MILP formulation
Vehicle routing problem
Dial-a-ride problem
Transfer

ABSTRACT

Automated vehicles are becoming a reality. Expectations are that AVs will ultimately transform personal mobility from privately owned assets to on-demand services. This transformation will enhance the possibility of sharing trips, leading to shared AVs (SAVs). The preeminent aim of this paper is to lay foundations for fast and efficient algorithms to be used in such new driving conditions. These algorithms must be able to solve Dial-a-Ride problems with transfers (DARPT). Hence, they should efficiently assign passengers to vehicles and routes while also: administering vehicles dispatch, determining convenient parking for idling vehicles and managing vehicle routing in real-time. In this paper, we develop two integer linear programming models (one in continuous time and one in discrete time) and their extensions to solve the DARPT. Our models take into account routing, service times, constraints on maximum route time-span, unserved requests, preferred arrival and departure time, nonconstant travel times, convenient parking while optimizing routing costs and quality of the service. The models are tested on instances based on Google Maps data by solving them with a commercial solver. The results of these tests are the starting point for validating the performance of forthcoming, ad hoc metaheuristics to be used in real-life sized scenarios.

1. Introduction

Automated vehicles (AVs) will reshape our transportation system. The opportunities and potential they offer will lead to the most significant transportation revolution since the introduction of the internal combustion engine (Spieser et al. (2014)). In fact, nowadays, cities are facing an increasing personal transportation demand combined with a growing population, while spatial resources remain static. Traditional solutions to congestion (roads expansion, added bus services, new subway lines, etc.) cannot mitigate the traffic escalation. In order to meet this expanding mobility demand, a new transportation mentality involving shared autonomous vehicles (SAVs) is likely to become the dominant mindset in the coming years (Fagnant and Kockelman (2014); International Transport Forum (2015)). Although this SAVs system has some disadvantages, such as high initial cost barriers (Fagnant and Kockelman (2015); Conceição et al. (2017)), loss of vehicles' status symbol (Correia and van Arem (2016)) and distrust from the public, this new mentality is accelerated by the growing number of environmentally conscious citizens and the rising popularity of on-demand ridesharing services, especially among young adults. In addition, thanks to the possible interactivity among SAVs and to a smart managing system on top, lower and lower congestion levels will be achieved (Spieser et al. (2014); International Transport Forum (2015); Liang et al. (2017)).

As pointed out by Spieser et al. (2014), while automated vehicle technology continues to move forward, less attention has been devoted to the logistics of effectively managing a fleet of potentially thousands of such vehicles. Accordingly, our paper aims to fill this gap to a certain degree by presenting exact formulations to solve the routing of SAVs. In particular, we demonstrate the benefits of transfers and ridesharing for small instances. The performance of forthcoming metaheuristics can be compared against these results. A more formal definition of 'ridesharing' and 'transfer' is presented in Section 2. Since organizing transfers is a meticulous and precise action that requires information on the position of all vehicles, we assume vehicles to be autonomous and to be managed by a centralized controller. Although these autonomous vehicles will most likely be electric, we do not take battery consumption into account in this research, since this will further complicate an already complex problem.

To simulate on-demand ridesharing systems, almost every aforementioned author (Fagnant and Kockelman (2014); Martinez et al. (2015); International Transport Forum (2015); Fagnant and Kockelman (2015); Liu et al. (2017)) used agent based simulation. In this paper, instead, we use exact optimization techniques to achieve optimal

* Corresponding author.
E-mail address: j.pierotti@tudelft.nl (J. Pierotti).

https://doi.org/10.1016/j.ejtl.2021.100037
Received 30 July 2020; Received in revised form 19 February 2021; Accepted 14 April 2021

solutions; however, this limits the ability to solve real-life sized instances due to high computational times.

The problem we aim to solve shares many characteristics with two traditional problems in the vehicle routing problem (VRP) literature: the (Splittable) Pick Up and Delivery Problem (PDP) and the Dial-A-Ride Problem (DARP). Since these are infamously known to be NP-Hard and difficult to approximate (Masson et al. (2014)), many authors focused on heuristic solution methods.

PDP shares most of the routing structure with our problem. The main difference lies in the fact that goods (typically letters or small parcels[1]) are carried instead of people. Hence, in the PDP, the quality of the time spent travelling is not considered. In addition, goods facilities may have time windows while, in general, there is no preferred arrival or departure time. Also, transfers are considered, but only in a few predetermined hubs. To solve the PDP with transfers, Rais et al. (2014) introduce an integer programming model and use standard branch-and-bound while Cortés et al. (2010) adopt branch-and-cut techniques. Also, Peng et al. (2019) developed a MILP formulation for the (selective) PDP with transfers and compared its solution with a particle swarm optimization metaheuristic. Given the complexity of the problem, Peng et al. (2019), Cortés et al. (2010) and Rais et al. (2014) solve instances with five, six and seven requests to optimality, respectively. Thangiah et al. (2007) combined a constructive heuristic with local optimization to quickly (under 5 s) solve the online version of the PDP. Danloup et al. (2018) tested two metaheuristics, namely large neighborhood search (LNS) and genetic algorithms (GA), showing a very performing implementation of the latter. Masson et al. (2013) developed an adaptive large neighborhood search (ALNS) algorithm with different destroy and repair methods, showing its performance on real-life data from the area of Nantes. Also, Petersen and Ropke (2011) solve the pickup and delivery problem with cross-docking opportunity (a variant of the PDP) using LNS. In addition, their algorithm was tested on real-life instances with sizes ranging from 500 to 1000 requests (but only one possible transfer node). All the aforementioned authors who solve the PDP considered only few nodes (hubs) as possible transfer nodes, while we consider all nodes as possible transfer nodes.

On the other hand, the DARP consists of designing routes for vehicles and schedules for users who specify pickup and delivery requests between origins and destinations. DARPs are a well studied class of problems and extensive reviews can be found in Agatz et al. (2012), Molenbruch et al. (2017) and Ho et al. (2018). In general, ridesharing is allowed but transfers are not considered (Cordeau and Laporte (2003)). Since people are transported, the quality of the travel time has to be considered. Most authors state that minimizing the routing costs or minimizing the time of the routes implies minimizing the loss of quality of the service. Although reasonable, we prefer a more explicit approach, as detailed in Section 3.2.2 and Section 4.1.1. Even though standard DARP problems do not take transfers into account, some papers do consider DARP with transfers (DARPT). Masson et al. (2014) used an adaptive strategy combined with a ruin and repair mechanism while Deleplanque and Quilliot (2013) developed a general insertion scheme. In these papers, fast heuristic conditions were introduced to check if a repaired route was feasible or not. Hou et al. (2016) developed a MILP formulation and a greedy heuristic to compare electric taxi usage between the nontransferable taxi-sharing and the transferable one. They show that, during rush hours, transfers could improve the number of served passengers and the shared travel distance by 22% and 37%, respectively. Nevertheless, transfers can only happen at recharging stations.

Moving outside the conventional boundaries of the DARP with transfers, Reinhardt et al. (2013) and Posada et al. (2017) developed multi modal frameworks. In these papers, transfers happen between different transport modes within an airport or between the public and the private transportation, respectively.

Our contribution is the development of two MILPs -one in continuous time and one in discrete time-to solve the DARP while considering:

- the possibility to transfer in all nodes of the network,
- routes with cycles such that a vehicle or a request can visit the same node several times,
- the absence of a depot location, or equivalently, all nodes are considered to be depots,
- nonconstant travel times,
- focus on quality of time: requests' preferred arrival and departure time and quality loss due to transfers and waiting times,
- attention to convenient parking.

As shown by van den Berg and van Essen (2019) and confirmed by our results (Section 5.3), either the continuous time models or the discrete one can perform better depending on the situation. In order to tie continuous and discrete variables in the continuous time model, we introduce the concept of 'move' (Section 3.1). To the best of our knowledge, cycles have been handled by creating dummy copies of nodes, but they have not been explicitly modelled in the literature. Also, cycles are very common in practical applications of people transportation, for instance, a taxi may visit the airport or the city center multiple times during a work shift. As shown in this paper, these features increase the complexity of the standard DARP but allow for more flexibility.

The remainder of this paper is organized as follows: the problem, our assumptions and the notation used are described in detail in Section 2. Section 3 and Section 4 depict the model in continuous time and in discrete time, respectively. Section 5 details performance and computational results and, finally, Section 6 concludes the paper.

2. Problem formulation

In this paper, we describe how to solve the Dial-a-Ride problem with transfers to optimality. The problem we consider is the following: given a set of requests R and a set of vehicles V, minimize the generalized cost Z (routing and quality of service), allowing transfers and ride sharing while considering time and vehicle limitations. Allowing transfers means that each passenger may be picked up by a vehicle, taken to a certain location, dropped off and picked up by another vehicle. This procedure may be repeated multiple times. Without loss of generality, we suppose the initial time to be zero.

The road network is modelled as a graph with node set N connected by a set of arcs A. For now, we assume that each vehicle can idle and each request can wait at every node which, in turn, are all possible transfer nodes. In Appendix B, we describe how the set of transfer nodes can be limited.

We consider the length l_{ij} of arc $(i,j) \in A$ with $i,j \in N$ to be known. The travel time of each arc is a function of the arc $(i,j) \in A$ and the time t at which a vehicle starts travelling it, i.e. $F(ijt)$. In this paper, we consider two cases: the case where $F(ijt)$ returns a parameter dependent only on arc $(i,j) \in A$ and the case where the travel time is depending on both arc $(i,j) \in A$ and time t. In Section 3.3.2 and Section 4, we describe how to derive a linear formulation even for nonconstant travel times (travel times depending on t).

Each request $r \in R$ is characterized by nine parameters. The first one, e^r, determines its earliest possible departure time; so, every request can be picked up only after time e^r. The second parameter, l^r, determines the time instant before which each request must reach its destination; otherwise, it is considered unserved. Every request $r \in R$ is also characterized by parameters pd^r and pa^r. These parameters state the preferred time instants for departure and arrival. It must hold that $e^r \le pd^r \le pa^r \le l^r$. Parameters e^r and l^r define a hard time window (on the request, not on the node) while pd^r and pa^r define a soft one. This

[1] since small packages are moved, the problem is often treated as capacity unconstrained.

newly introduced type of time limitation differs from standard hard time windows, which are imposed on a node and not on a client, as well as from maximum ride time, which are imposed on a client but do not consider departure and arrival time *per se* but only their difference. The maximum number of transfers allowed for request $r \in R$ is given by b^r and the party size of the same request is given by q^r. The origin and destination of request $r \in R$ is given by $o^r \in N$ and $d^r \in N$, respectively.

Each vehicle $v \in V$ is described by an initial position $o^v \in N$ and a capacity q^v. $\alpha, \beta, \gamma_1, \gamma_2, \gamma_3, \gamma_4, \eta$ and E are cost parameters detailed in Section 5.2, while B is a sufficiently large number. We name ε the minimum travel time among any arc $(i,j) \in A$ (i.e. $\min_{ij \in A, t \geq 0} F(ijt)$) and we let T_{Max} be equal to the maximum latest arrival time l^r over all requests $r \in R$ (i.e. $\max_{r \in R} l^r$).

3. Continuous time model

In this section, we detail the concept of a 'move' (Section 3.1), the core continuous time model (Section 3.2) and its extension (Section 3.3). The extended model fulfills the same aim as the core model (i.e. solving the DARPT), but also includes additional features such as service times and nonconstant travel times. These features are added to have a closer resemblance to real-life. Both the continuous time and discrete time model rely on the idea of tracking requests' flows (from their origin to their destinations) and forcing vehicles' flows to overlap and be paired with them (more details in Section 3.2.6 and Section 4.1.3). We refer to the formulations based on this idea as *flow formulations*.

3.1. Moves

While designing routes involves discrete variables, adopting a continuous time model implies having variables in continuous time. Since discrete and continuous variables influence each other and have to be considered simultaneously, we introduce the 'move' concept. A 'move' refers to the act of travelling an arc. In fact, we associate each request $r \in R$ with a set $M^r = \{1, 2, 3, .., \mathscr{M}^r\}$. The cardinality $\mathscr{M}^r$ of this set is bigger than the maximum number of arcs request $r \in R$ can travel given its time limits (from e^r to l^r). In particular, given the time limitations on each request and a non degenerative graph (i.e. each travel time is strictly greater than zero), there is a bound on the maximum number of arcs a request can possibly travel. Equivalently, there is a bound on the maximum number of arcs that a vehicle can travel; so, each vehicle $v \in V$ is also associated with a set of moves $M^v = \{1, 2, 3, .., \mathscr{M}^v\}$. The cardinalities of sets M^r and M^v are given by $\mathscr{M}^r = \left\lfloor \frac{l^r - e^r}{\varepsilon} \right\rfloor$ and $\mathscr{M}^v = \left\lfloor \frac{T_{Max}}{\varepsilon} \right\rfloor$, respectively. Fig. 1 graphically shows how these moves are counted. The text next to each arc indicates who is travelling that arc and at which move. The figure clearly shows that vehicles and requests can travel the same arc at different moves.

All the parameters described in Section 2 and in this section are summarized in Table 1.

Table 1
Sets, parameters and function.

R	Set of requests.
e^r	earliest departure time for request $r \in R$.
pd^r	preferred departure time for request $r \in R$.
pa^r	preferred arrival time for request $r \in R$.
l^r	latest arrival time for request $r \in R$.
b^r	maximum number of transfers allowed per request r.
q^r	party size of request $r \in R$ (how many people in request $r \in R$).
o^r	origin of request $r \in R$.
d^r	destination of request $r \in R$.
M^r	set of all the possible moves for request $r \in R$.
$\mathscr{M}^r$	cardinality of set M^r for request $r \in R$.
V	set of vehicles.
o^v	origin of vehicle $v \in V$.
q^v	capacity of vehicle $v \in V$.
M^v	set of all the possible moves for vehicle $v \in V$.
$\mathscr{M}^v$	cardinality of set $M^v, v \in V$.
N	set of nodes.
A	set of all arcs.
l_{ij}	length of arc $(i,j) \in A$.
$F(ijt)$	function that returns the travel time along arc $(i,j) \in A$ at time t.
T_{Max}	time instant after which no requests can be delivered, i.e. $T_{Max} \geq l^r$, $\forall r \in R$.
ε	shortest travel time in the network, i.e. $\min_{(i,j) \in A, t \geq 0} F(ijt)$.
$\alpha, \beta, \gamma_1, \gamma_2, \mu_1, \mu_2, \mu_3, \mu_4, \eta, E$	cost coefficients.
B	parameter with high value.
Z	generalized cost.

3.2. Core model

After introducing the concept of 'move', we dedicate this section to the description of the core continuous model. This core model takes into account routing, timing, pairing, capacity constraints, constraints on the maximum number of possible transfers and the possibility of unserved requests while minimizing a trade off between routing costs and the loss of the quality of the service.

3.2.1. Variables

In this section, we describe the variables used in the model. To model the routing, we employ binary variables x^r_{ijm} which assume value one when request $r \in R$ travels arc $(i,j) \in A$ in move $m \in M^r$ and zero otherwise. If considered in the order of the moves, these variables describe the route of request $r \in R$. Next to the routing, variables describing the chronological framework are needed; for this, we introduce continuous variables t^r_m and w^r_m for request $r \in R$ and move $m \in \{0\} \cup M^r$. Since m defines the move from one node to another, t^r_m assumes the value of the time instant at which request $r \in R$ arrives in a node after move $m \in M^r$, while w^r_m assumes the value of how much time

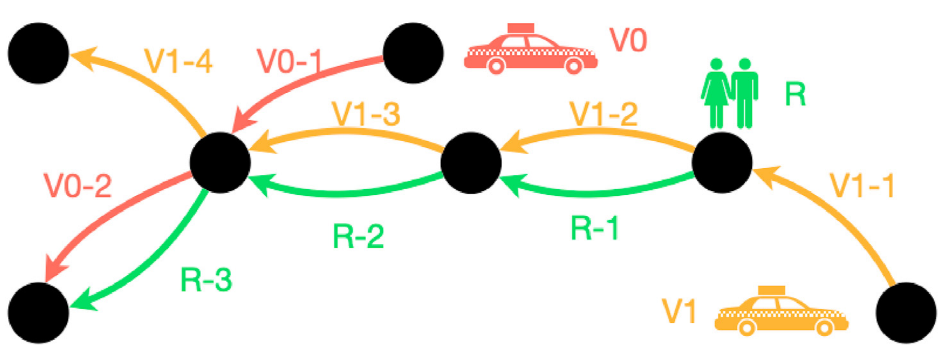


Fig. 1. Example of moves. Colors refer to who is travelling that arc (red for vehicle $V0$, yellow for vehicle $V1$ and green for request R). The text next to each arc indicates who is travelling that arc and at which move. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

request $r \in R$ waits after move $m \in M^r$. In the timing and waiting variables (t_m^r and w_m^r), the additional move $\{0\}$ is considered to determine the initial departure time which occurs before the first move. Fig. 2 depicts the relation between x_{ijm}^r, t_m^r and w_m^r. In this figure, we analyze the first two moves of the route travelled by request $r \in R$. We suppose the origin o^r to be in node $i \in N$. The black arrows illustrate the moves; in particular, the first move is from node $i \in N$ to node $j \in N$ while the second one is from node $j \in N$ to node $k \in N$. The red arrows are associated with the timing variables t_m^r and indicate which node the arrival time values refers to. The green boxes relate to the waiting times w_m^r and are depicted next to the nodes they are assigned to.

Equivalent variables for the vehicles are needed. For routing purposes, we introduce binary variables y_{ijm}^v which are one when vehicle $v \in V$ travels arc $(i,j) \in A$ at move $m \in M^v$ and zero otherwise. In addition, continuous variables t_m^v and w_m^v are used to characterize the timing of vehicle $v \in V$ for move $m \in \{0\} \cup M^v$.

Naming c_r the actual arrival time of request $r \in R$ to its destination d^r, we define continuous variables c_r^+ and c_r^- such that $c_r^+ = \max(c_r, pa^r)$ and $c_r^- = \min(c_r, pa^r)$. Similar considerations apply for continuous variables d_r^+ and d_r^- which, in turn, are constrained such that $d_r^+ = \max(z^r, pd^r)$ and $d_r^- = \min(z^r, pd^r)$, where z^r is the actual departure time of request $r \in R$ from its origin o^r. This is useful to penalize late or early arrival and departure times, as described in Section 3.2.2.

Binary variables a_{rv} are one when request $r \in R$ is paired with vehicle $v \in V$ during its route and zero otherwise. In addition, binary variables $p_{rm_r vm_v}$ are one when request $r \in R$ during move $m_r \in M^r$ is paired with vehicle $v \in V$ during move $m_v \in M^v$ and zero otherwise. The behaviour of these variables is explained in detail in Section 3.2.6. Finally, we introduce binary variable u^r which assumes value one when request $r \in R$ is unserved and zero otherwise. Table 2 reports all the variables previously described.

3.2.2. Objective function

The objective function aims at minimizing the generalized cost Z. This cost is set equal to the sum of the following eight terms. The first term indicates the travel cost of a solution which is given by:

$$\alpha \sum_{v \in V} \sum_{(i,j) \in A} \sum_{m \in M^v} y_{ijm}^v l_{ij}.$$

To penalize the time travelled by each passenger inside a vehicle (which is computed as arrival time minus departure time and waiting time), we introduce the second term:

$$\beta \sum_{r \in R} q^r \left(t_{\mathscr{M}^r}^r - \sum_{m \in \{0\} \cup M^r} w_m^r - t_0^r \right).$$

In this second term, $t_{\mathscr{M}^r}^r$ assumes the value of the time instant at which a request arrives at its destination. This is explained in detail in Section 3.2.4.

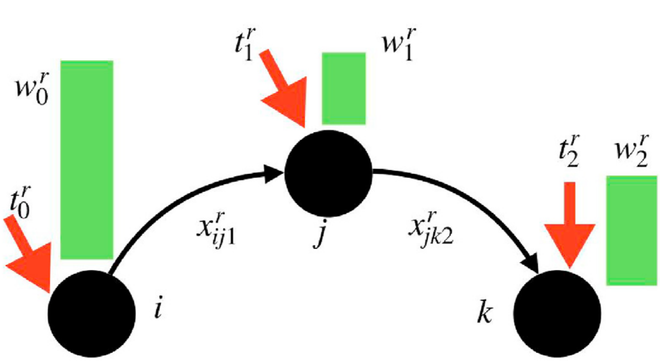


Fig. 2. Example of routing (black arrows), timing (red arrows) and waiting variables (green boxes) for the first two moves of request $r \in R$. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

Table 2
Variables continuous time model.

t_m^r	time variable indicating when request $r \in R$ arrives at a node after move $m \in \{0\} \cup M^r$.
t_m^v	time variable indicating when vehicle $v \in V$ arrives at a node after move $m \in \{0\} \cup M^v$.
w_m^r	waiting variable indicating how much time request $r \in R$ waits after move $m \in \{0\} \cup M^r$.
w_m^v	waiting variable indicating how much time vehicle $v \in V$ waits after move $m \in \{0\} \cup M^v$.
x_{ijm}^r	binary variable which is one if request $r \in R$ travels arc $(i,j) \in A$ at move $m \in M^r$, zero otherwise.
y_{ijm}^v	binary variable which is one if vehicle $v \in V$ travels arc $(i,j) \in A$ at move $m \in M^v$, zero otherwise.
c_r^+	variable indicating late arrival for request $r \in R$.
c_r^-	variable indicating early arrival for request $r \in R$.
d_r^+	variable indicating late departure for request $r \in R$.
d_r^-	variable indicating early departure for request $r \in R$.
a_{rv}	binary variable which is one if request $r \in R$ is paired with vehicle $v \in V$, zero otherwise.
$p_{rm_r vm_v}$	binary variable which is one if request $r \in R$ at move $m_r \in M^r$ is paired with vehicle $v \in V$ at move $m_v \in M^v$, zero otherwise.
u^r	binary variable which is one if request $r \in R$ is unserved, zero otherwise.

The third term handles early and late departure while the fourth term determines the penalty for early and late arrival:

$$\mu_1 \sum_{r \in R} (pd^r - d_r^-) + \mu_2 \sum_{r \in R} (d_r^+ - pd^r),$$

$$\mu_3 \sum_{r \in R} (c_r^+ - pa^r) + \mu_4 \sum_{r \in R} (pa^r - c_r^-).$$

We assign a penalty related to the loss of quality every time there is a transfer. This is taken into account by the fifth term:

$$\eta \sum_{r \in R} \sum_{v \in V} a_{rv} q^r.$$

In order to penalize how much time passengers wait at transfer nodes, we add the following sixth term:

$$\gamma_1 \sum_{r \in R} \sum_{m \in M^r} q^r w_m^r.$$

In this sixth term, the first move (move number zero) is excluded because, for a request, waiting at its origin node is already penalized as an early or late departure.

The seventh term determines parking costs, which are considered proportional to the parking time:

$$\gamma_2 \sum_{v \in V} \sum_{m \in \{0\} \cup M^v} w_m^v.$$

Finally, the last term penalizes the unserved requests:

$$+E \sum_{r \in R} u^r q^r.$$

Hence, the generalized cost Z is given by:

$$\begin{aligned} Z = {} & \alpha \sum_{v \in V} \sum_{(i,j) \in A} \sum_{m \in M^v} y_{ijm}^v l_{ij} + \beta \sum_{r \in R} q^r \left(t_{\mathscr{M}^r}^r - \sum_{m \in \{0\} \cup M^r} w_m^r - t_0^r \right) \\ & + \mu_1 \sum_{r \in R} (pd^r - d_r^-) + \mu_2 \sum_{r \in R} (d_r^+ - pd^r) + \mu_3 \sum_{r \in R} (c_r^+ - pa^r) \\ & + \mu_4 \sum_{r \in R} (pa^r - c_r^-) + \eta \sum_{r \in R} \sum_{v \in V} a_{rv} q^r + \gamma_1 \sum_{r \in R} \sum_{m \in M^r} q^r w_m^r \\ & + \gamma_2 \sum_{v \in V} \sum_{m \in \{0\} \cup M^v} w_m^v + E \sum_{r \in R} u^r q^r. \end{aligned} \tag{1}$$

Although this objective function is quite elaborate, it approximates real-life.

3.2.3. Routing constraints

Firstly, to model the routing, we have to impose that, at each move, at most one arc can be chosen. This is provided by constraints (2).

$$\sum_{(i,j)\in A} x^r_{ijm} \le 1, \forall r \in R, \forall m \in M^r \tag{2}$$

Also, we have to ensure that each request is either unserved (i.e. $u^r = 1$) or starts at its origin, possibly travel through some nodes, and finally ends at its destination. This is ensured by constraints (3)–(5). In particular, constrains (3) enforce that, if a request moves (i.e. is served), it has to start from its origin. Constraints (4) establish flow conservation in each node while constraints (5) impose that request $r \in R$ has either to arrive at its destination or it is unserved.

$$\sum_{(o^r,j)\in A} x^r_{o^r j1} \ge \sum_{(i,j)\in A} x^r_{ij1}, \forall r \in R, \tag{3}$$

$$\sum_{(i,j)\in A} x^r_{ijm} = \sum_{(j,k)\in A} x^r_{jk(m+1)}, \forall r \in R, \forall m \in M^r \setminus \{\mathscr{M}^r\}, j \ne d^r, \tag{4}$$

$$u^r + \sum_{m_r\in M^r} \sum_{(i,d^r)\in A} x^r_{id^r m_r} = 1, \forall r \in R \tag{5}$$

Similar considerations hold for the vehicle flow. Constraints (6) restrict each vehicle to select at most one arc per move while constraints (7) force a moving vehicle to start from its origin. Constraints (8) ensures flow conservation in all nodes. In fact, this set of constraints imposes that, if a vehicle uses exactly $\widehat{m} \le \mathscr{M}^v$ arcs to reach its destination, it holds that for each $m \le \widehat{m}$, $\sum_{(i,j)\in A} y^v_{ijm} = 1$ and that after the $\widehat{m}_{th}$ move $\sum_{(i,j)\in A} y^v_{ijm} = 0$. The last move is not considered because it has no following move ($m+1$). Vehicles have to obey constraints similar to the ones of the requests, but have no defined destination node; hence, there is no associated constraint.

$$\sum_{(i,j)\in A} y^v_{ijm} \le 1, \forall v \in V, \forall m \in M^v \tag{6}$$

$$\sum_{(o^v,j)\in A} y^v_{o^v j1} \ge \sum_{(i,j)\in A} y^v_{ij1}, \forall v \in V \tag{7}$$

$$\sum_{(i,j)\in A} y^v_{ijm} \ge \sum_{(j,k)\in A} y^v_{jk(m+1)}, \forall v \in V, \forall m \in M^v \setminus \{\mathscr{M}^v\}, \forall j \in N \tag{8}$$

3.2.4. Timing constraints

Without loss of generality, we assume time to start at zero. Firstly, we ensure, through constraints (9) and (10), timing and waiting variables to be positive.

$$t^r_m \ge 0, \forall r \in R, \forall m \in \{0\} \cup M^r \tag{9}$$

$$w^r_m \ge 0, \forall r \in R, \forall m \in \{0\} \cup M^r \tag{10}$$

Then, we impose chronological timing. With chronological timing, we mean that the arrival time at any node (but the first) is the arrival time at the previous node plus the waiting time at the previous node plus the travel time. This is ensured by constraints (11). Clearly, if $F(ijt)$ is a time independent parameter, constraints (11) are linear. Nonetheless, in Section 3.3.2, we derive a linear formulation for the case where the travel time depends on the departure time. If no arc is chosen, then $\sum_{(i,j)\in A} x^r_{ijm}$ is zero, hence $t^r_{m+1} = t^r_m + w^r_m$.

$$t^r_{m+1} = t^r_m + w^r_m + \sum_{(i,j)\in A} x^r_{ijm} F\big(ij\big(t^r_m + w^r_m\big)\big), \forall r \in R, \forall m \in \{0\} \cup M^r \setminus \{\mathscr{M}^r\} \tag{11}$$

Also, we force the initial time of request $r \in R$ to be its earliest time instant (constraints (12)), and we force its last time instant to be smaller than its latest arrival time (constraints (13)).

$$t^r_0 = e^r, \forall r \in R \tag{12}$$

$$t^r_{\mathscr{M}^r} \le l^r, \forall r \in R \tag{13}$$

To not have conveniently large waiting times at the end of the route to better fit time preferences, we impose constraints (14). In fact, when no arc is chosen for a certain move, i.e. request $r \in R$ has reached its destination, we have no waiting time. This means that $t^r_m = t^r_{\widehat{m}}$ for $m \ge \widehat{m}$, with $\widehat{m}$ the move with which request $r \in R$ has reached its destination.

$$w^r_m \le B \sum_{(i,j)\in A} x^r_{ijm}, \forall r \in R, \forall m \in M^r \tag{14}$$

Constraints (13) do not imply that we have to reach the destination node at the last move. Rather, they say that if a request reaches its destination at move $\widehat{m}$, then $t^r_m = t^r_{\widehat{m}}$ for $\widehat{m} \le m \le \mathscr{M}^r$. This is ensured by the absence of waiting times once the destination is reached (constraints 14).

For vehicles, we duplicate the equivalent of the timing constraints with small adjustments; in particular, we impose:

$$t^v_m \ge 0, \forall v \in V, \forall m \in \{0\} \cup M^v, \tag{15}$$

$$w^v_m \ge 0, \forall v \in V, \forall m \in \{0\} \cup M^v, \tag{16}$$

$$t^v_{m+1} = t^v_m + w^v_m + \sum_{(i,j)\in A} y^v_{ijm} F\big(ij\big(t^v_m + w^v_m\big)\big), \forall v \in V, \forall m \in \{0\} \cup M^v \setminus \{\mathscr{M}^v\}, \tag{17}$$

$$t^v_0 = 0, \forall v \in V, \tag{18}$$

$$t^v_{\mathscr{M}^v} \le T_{Max}, \forall v \in V, \tag{19}$$

$$w^v_m \le B \sum_{(i,j)\in A} y^v_{ijm}, \forall v \in V, \forall m \in M^r. \tag{20}$$

3.2.5. Departure and arrival times constraints

We want variable d^+_r to assume the maximum of the preferred departure time pd^r and the actual departure time $t^r_0 + w^r_0$ of request $r \in R$ and d^-_r to assume the minimum of these two. Hence, we adopt the following constraints:

$$d^+_r \ge pd^r, \forall r \in R \tag{21}$$

$$d^+_r \ge t^r_0 + w^r_0, \forall r \in R \tag{22}$$

$$d^-_r \le pd^r, \forall r \in R \tag{23}$$

$$d^-_r \le t^r_0 + w^r_0, \forall r \in R \tag{24}$$

Similar constraints hold for the arrival time.

$$c^+_r \ge pa^r, \forall r \in R \tag{25}$$

$$c^+_r \ge t^r_{\mathscr{M}^r}, \forall r \in R \tag{26}$$

$$c^-_r \le pa^r, \forall r \in R \tag{27}$$

$$c^-_r \le t^r_{\mathscr{M}^r}, \forall r \in R \tag{28}$$

Given the previous constraints and the composition of the objective function, the departure and arrival time will be forced to be as close as possible to pd^r and pa^r.

3.2.6. Pairing

In this section, we explain how to pair vehicles and requests. We use moves to discretize routes even though the timing is expressed in continuous variables. We use binary variable $p_{rm_rvm_v}$ to pair couple [request $r \in R$ - request move $m_r \in M^r$], with couple [vehicle $v \in V$- vehicle move $m_v \in M^v$]. On one hand, we impose that each request can be paired to at most one vehicle per move in constraints (29); on the other hand, a vehicle can be paired simultaneously with multiple requests as long as this does not violate its capacity, as expressed in constraints (30).

$$\sum_{v \in V} \sum_{m_v \in M^v} p_{rm_rvm_v} \le 1, \forall r \in R, m_r \in M^r \tag{29}$$

$$\sum_{r \in R} \sum_{m_r \in M^r} p_{rm_rvm_v} q^r \le q^v, \forall v \in V, m_v \in M^v \tag{30}$$

As long as a request and a vehicle are paired, they have to travel the same arcs (constraints (31) and (32)) at the same time (constraints (33) and (34)). Constraints (31)–(34) are the linearization of:

$$\sum_{(i,j) \in A} x^r_{ijm_r} y^v_{ijm_v} \ge p_{rm_rvm_v}, \forall r \in R, \forall m_r \in M^r, \forall v \in V, \forall m_v \in M^v$$

and

$$p_{rm_rvm_v}\left(t^v_{m_v} - t^r_{m_r}\right) = 0, \forall v \in V, \forall r \in R, \forall m_r \in M^r, \forall m_v \in M^v$$

which are always true when $p_{rm_rvm_v} = 0$. The inequality in the first nonlinear set of constraints is due to the fact that, by chance, a vehicle and a request may travel the same arc while not being paired together.

$$x^r_{ijm_r} \le y^v_{ijm_v} + \left(1 - p_{rm_rvm_v}\right), \forall v \in V, \forall r \in R, \forall (i,j) \in A, \forall m_r \in M^r, \forall m_v \in M^v \tag{31}$$

$$x^r_{ijm_r} \ge y^v_{ijm_v} - \left(1 - p_{rm_rvm_v}\right), \forall v \in V, \forall r \in R, \forall (i,j) \in A, \forall m_r \in M^r, \forall m_v \in M^v \tag{32}$$

$$t^r_{m_r} \le t^v_{m_v} + \left(1 - p_{rm_rvm_v}\right)B, \forall v \in V, \forall r \in R, \forall m_r \in M^r, \forall m_v \in M^v \tag{33}$$

$$t^r_{m_r} \ge t^v_{m_v} - \left(1 - p_{rm_rvm_v}\right)B, \forall v \in V, \forall r \in R, \forall m_r \in M^r, \forall m_v \in M^v \tag{34}$$

Constraints (33) and (34) (in combination with the timing constraints (17)) impose that, if a vehicle is paired to more than one request in $i \in N$, the arrival time in $j \in N$ of the vehicle and of all the paired requests is equal to the arrival time of the request that arrived the latest in $i \in N$ plus its waiting time plus the travel time through $(i,j) \in A$.

In addition, if a request is not paired with any vehicle, it cannot move (constraints (35)).

$$\sum_{(i,j) \in A} x^r_{ijm_r} \le \sum_{v \in V} \sum_{m_v \in M^v} p_{rm_rvm_v}, \forall r \in R, \forall m_r \in M^r \tag{35}$$

Finally, we impose an upperbound on the maximum number of transfers a request can experience (constraints (36) and (37)). Variable a_{rv} captures if request $r \in R$ has ever been paired to vehicle $v \in V$. The number of transfers a request can encounter is the number of vehicles the request has been paired with minus the first one.

$$Ba_{rv} \ge \sum_{m_r \in M^r} \sum_{m_v \in M^v} p_{rm_rvm_v}, \forall v \in V, \forall r \in R \tag{36}$$

$$\sum_{v \in V} a_{rv} - 1 \le b^r, \forall r \in R \tag{37}$$

3.2.7. Double pick up

There is an extremely unlikely case where the model does not see a transfer. This eventuality never happens in our instances and rarely can happen in practice; nevertheless, for the sake of completeness, we explain the possible problem throughout an example (depicted in Fig. 3). Suppose that we have 5 time instants in chronological order, $t_1 < t_2 < t_3 < t_4 < t_5$, and that vehicle $v \in V$ and request $r \in R$ are paired at time t_1. Then, at time t_2 the request is dropped off at node $i \in N$ while the vehicle continues its route reaching node $j \in N$ at time t_3. Finally, the same vehicle comes back to node i at time t_4 to pick up again request $r \in R$ and brings it to node $k \in N$ at time t_5. This second pick up is not counted in the model. In general, we can say that if a particular vehicle, for multiple times, picks up and drops off the very same request consecutively, not all the resulting transfers are properly counted. Please note that if two different vehicles sequentially pick up the same request, the transfer is correctly accounted for by the model (as we show in the example depicted in Fig. 8b). This double pick up by the same vehicle can happen only when there are large time windows and tight capacity constraints at the same time. In fact, the complete route of a new request should fit inside the route of an already existing one. Time limitations make this possibility unlikely. Given the extreme rarity of this situation, we do not consider this eventuality.

3.3. Model extension

In this section, we show how to extend the model such that it considers people dependent service times (time for get-in operations). As motivated by Ichoua et al. (2003), Donati et al. (2008) and Schmid and Doerner (2010), we also consider nonconstant travel times. Both features are added to have a closer resemblance to real-life.

3.3.1. People dependent service time

In this section, we explain how to consider different service times for different requests (people dependent service time). As service time, we consider the time needed to get in a vehicle. Also, we assume each request $r \in R$ to be picked up by a particular vehicle $v \in V$ at most once. Each request is considered to have a known service time named s^r. We introduce binary variable g^r_m which assumes value one if request $r \in R$ experiences a get in operation at move $m \in M^r$ and zero otherwise. To ensure this, we impose the following three constraints: constraints (38) for the first move of a request, constraints (39) for the first move of a

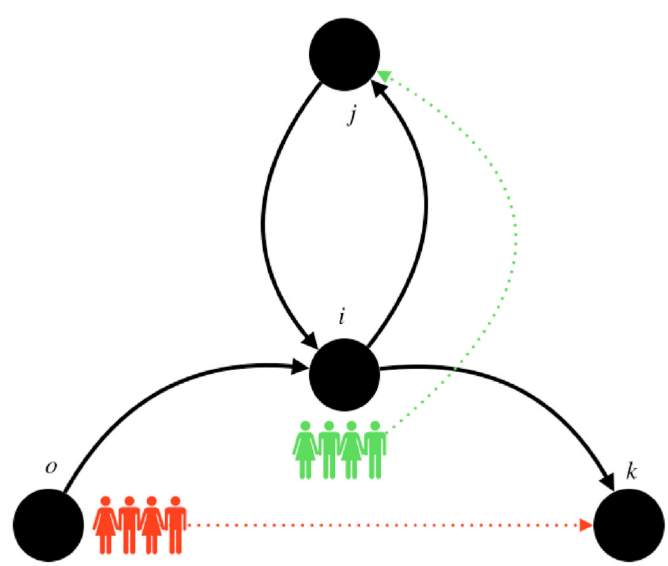


Fig. 3. Example of a double pick up by the same vehicle. The black arrows represent the route of the vehicle (o,i,j,i,k). The red request has origin o and destination k while the green request has origin i and destination j. The combined party sizes of the two requests exceeds the capacity of the vehicle. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

vehicle and constraints (40) for all other moves of requests and vehicles.

$$g_1^r \geq \sum_{v\in V}\sum_{m_v\in M^v} p_{r1vm_v}, \forall r \in R \tag{38}$$

$$g_{m_r}^r \geq \sum_{v\in V} p_{rm_rv1}, \forall r \in R, \forall m_r \in M^r \tag{39}$$

$$g_{m_r}^r \geq p_{rm_rvm_v} - p_{rm_r-1vm_v-1}, \forall v \in V, \forall m_v \in M^v \setminus \{1\}, \forall r \in R, \forall m_r \in M^r \setminus \{1\} \tag{40}$$

Constraints (40) exploit that, when a request is picked up, the vehicle and the request were not paired at the previous move (hence, $p_{rm_r-1vm_v-1} = 0$) while, at the current move, they are paired (hence, $p_{rm_rvm_v} = 1$). Thus, the difference between the two values is one. The corresponding timing constraints (11) are then modified to:

$$t_{m+1}^r = t_m^r + w_m^r + g_m^r s^r + \sum_{(i,j)\in A} x_{ijm}^r F\left(ij\left(t_m^r + w_m^r\right)\right), \forall r\in R, \forall m \in \{0\}\cup M^r \setminus \{\mathscr{M}^r\}. \tag{41}$$

We assume that, when vehicle $v\in V$ picks up two requests simultaneously from the same node, only the largest service time is considered. In Appendix A.1, we show a different formulation for the case when service times add up.

To model this for vehicles, we introduce binary variables h_{vmr} which assumes value one if and only if two conditions apply. The first condition is that vehicle $v \in V$ at move $m \in M^v$ picks up request $r \in R$ (and maybe some others). The second condition is that, among all the requests picked up at move $m \in M^v$ by vehicle $v \in V$, request $r \in R$ has the highest service time. To ensure that h_{vmr} can be one for only one request at move $m \in M^v$ of vehicle $v \in V$, we impose constraints (42).

$$\sum_{r\in R} h_{vmr} \leq 1, \forall v \in V, \forall m \in M^v \tag{42}$$

To ensure the selection of the request with the highest service time, we impose the following three constraints: constraints (43) for the first move of a vehicle, constraints (44) for the first move of a request and constraints (45) for all other moves of requests and vehicles.

$$\sum_{r\in R} h_{v1r} s^r \geq p_{rm_rv1} s^r, \forall v \in V, \forall r \in R, \forall m_r \in M^r. \tag{43}$$

$$\sum_{r\in R} h_{vm_vr} s^r \geq p_{r1vm_v} s^r, \forall v \in V, \forall m_v \in M^v, \forall r \in R. \tag{44}$$

$$\sum_{r\in R} h_{vm_vr} s^r \geq \left(p_{rm_rvm_v} - p_{rm_r-1vm_v-1}\right) s^r, \forall v\in V, \forall m_v \in M^v \setminus \{1\}, \forall r\in R, \forall m_r \in M^r \setminus \{1\}. \tag{45}$$

For vehicle $v\in V$ at move $m_v\in M^v$, constraints (43)–(45) force binary variables h_{vmr} to be one for request $r\in R$ such that its related service time s^r is greater than or equal to the service times that the vehicle is experiencing. Then, we impose through constraints (46) and (47) a tight upper bound. In fact, constraints (46) allow variable h_{vmr} to assume value one only if request $r\in R$ was not paired to vehicle $v\in V$ at the previous move. Constraints (47) force h_{vmr} to be zero if request $r\in R$ is not paired with vehicle $v\in V$ at move $m\in M^v$.

$$h_{vm_vr} \leq 1 - \sum_{m_r\in M^r} p_{rm_rvm_v-1}, \forall v \in V, \forall m_v \in M^v \setminus \{1\}, \forall r \in R \tag{46}$$

$$h_{vm_vr} \leq \sum_{m_r\in M^r} p_{rm_rvm_v}, \forall v \in V, \forall m_v \in M^v, \forall r \in R \tag{47}$$

Finally, we change constraints (17) into:

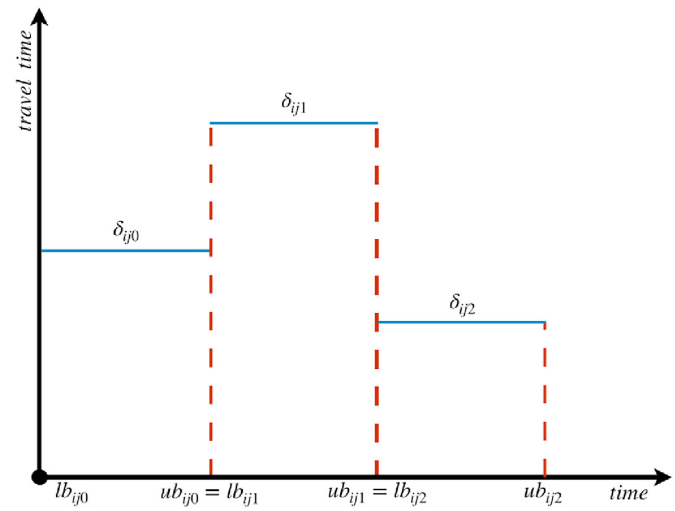


Fig. 4. Example of three time slots.

$$t_{m+1}^v = t_m^v + w_m^v + \sum_{r\in R} s^r h_{vmr} + \sum_{(i,j)\in A} y_{ijm}^v F\left(ij\left(t_m^v + w_m^v\right)\right), \forall v\in V, \forall m\in\{0\}\cup M^v \setminus \{\mathscr{M}^v\}. \tag{48}$$

3.3.2. Nonconstant travel time

So far, we have considered the generic function $F(ijt)$. In this section, we show how to linearize this function. A similar approach to deal with nonconstant travel times can be found in Malandraki and Daskin (1992). The main difference is that Malandraki and Daskin (1992) developed a method for the TSP; hence, for each node, exactly one incoming and one outgoing arc is travelled. With our formulation instead, for each node, zero, one, or more incoming and outgoing arcs can be travelled.

Firstly, when nonconstant travel times are taken into account, we need to ensure the same starting time for each pairing.[2] Hence, we add constraints (49) and (50). These constraints explicitly force the starting time of a pairing to be the same.

$$t_{m_r-1}^r + w_{m_r-1}^r \leq t_{m_v-1}^v + w_{m_v-1}^v + \left(1 - p_{rm_rvm_v}\right)B, \forall v\in V, \forall r\in R, \forall m_r \in M^r, \forall m_v \in M^v \tag{49}$$

$$t_{m_r-1}^r + w_{m_r-1}^r \geq t_{m_v-1}^v + w_{m_v-1}^v - \left(1 - p_{rm_rvm_v}\right)B, \forall v\in V, \forall r\in R, \forall m_r \in M^r, \forall m_v \in M^v \tag{50}$$

Since variables t_m^r and w_m^r are defined for $m\in\{0\}\cup M^r$, constraints (49) can range for all $m_r \in M^r$ even though they refer to time and waiting variable at the previous move. Similar considerations hold for constraints (50) and variables t_m^v and w_m^v. Secondly, we define TS as the set of time slots in which we can have different travel times (e.g. three time slots in Fig. 4).

In order to have a finite number of time slots, the function $F(ijt)$ must be piece-wise constant. If not, it is still possible to approximate $F(ijt)$ to a piece-wise discrete function and use its approximation. We introduce binary variables $TT_{ijm_rk}^r$ and $TT_{ijm_vk}^v$ which are one when arc $(i,j) \in A$ is chosen for move $m_r \in M^r$ of request $r \in R$ and for move $m_v \in M^v$ of vehicle $v \in V$, respectively, in time slot $k \in TS$, and zero otherwise. Parameters δ_{ijk} represent the amount of time needed to travel arc $(i,j) \in A$ in the time slot $k \in TS$. Each time slot $k \in TS$ is defined by a lower and an upper bound - namely, lb_{ijk} and ub_{ijk}, respectively - such that $ub_{ijk-1} =$

[2] For example, given time-dependent travel time $T(t)$ such that $T(t_1) = 3, T(t_2) = 2$ and $t_2 = t_1 + 1$, then different starting times can lead to the same arrival time.

lb_{ijk}. Subsequently, we need to constrain variables TT^r_{ijmk} and TT^v_{ijmk} to assume value one if and only if, at move $m \in M^v$ or $m \in M^r$, respectively, arc $(i,j) \in A$ is chosen at time t which falls into time slot $k \in TS$. To ensure this, we impose constraints (51), (52) and (53) for the requests and constraints (54), (55) and (56) for the vehicles.

$$\sum_{k \in TS} TT^r_{ijmk} = x^r_{ijm}, \forall r \in R, \forall m \in M^r, \forall (i,j) \in A \tag{51}$$

$$\sum_{(i,j) \in A} TT^r_{ijmk} lb_{ijk} \leq t^r_m + w^r_m + g^r_m s^r, \forall r \in R, \forall m \in M^r, \forall k \in TS \tag{52}$$

$$t^r_m + w^r_m + g^r_m s^r \leq \sum_{(i,j) \in A} TT^r_{ijmk} ub_{ijk} + B\left(1 - \sum_{(i,j) \in A} TT^r_{ijmk}\right), \forall r \in R, \forall m \in M^r, \forall k \in TS \tag{53}$$

$$\sum_{k \in TS} TT^v_{ijmk} = y^v_{ijm}, \forall v \in V, \forall m \in M^v, \forall (i,j) \in A \tag{54}$$

$$\sum_{(i,j) \in A} TT^v_{ijmk} lb_{ijk} \leq t^v_m + w^v_m + \sum_{r \in R} s^r h_{vmr}, \forall v \in V, \forall m \in M^v, \forall k \in TS \tag{55}$$

$$t^v_m + w^v_m + \sum_{r \in R} s^r h_{vmr} \leq \sum_{(i,j) \in A} TT^v_{ijmk} ub_{ijk} + B\left(1 - \sum_{(i,j) \in A} TT^v_{ijmk}\right), \forall v \in V, \forall m \in M^v, \forall k \in TS \tag{56}$$

Variables $TT^r_{ijm_rk}$ and $TT^v_{ijm_vk}$ can fully substitute variables $x^r_{ijm_r}$ and $y^v_{ijm_v}$, as we can clearly see from the equality constraints (51) and (54).

Finally, we change constraints (41) to:

$$t^r_{m+1} = t^r_m + w^r_m + g^r_m s^r + \sum_{k \in TS} \sum_{(i,j) \in A} TT^r_{ijmk} \delta_{ijk}, \forall r \in R, \forall m \in M^r \setminus \{\mathscr{M}^r\}, \tag{57}$$

and constraints (48) to:

$$t^v_{m+1} = t^v_m + w^v_m + \sum_{r \in R} s^r h_{vmr} + \sum_{k \in TS} \sum_{(i,j) \in A} TT^v_{ijmk} \delta_{ijk}, \forall v \in V, \forall m \in M^v \setminus \{\mathscr{M}^v\}. \tag{58}$$

In this way, we obtain a linear formulation which is able to consider nonconstant travel times.

4. Discrete time model

In this section, we describe the core discrete time model (Section 4.1) and its extension (Section 4.2). We assume the discretization step to be one time unit; if not, every time related equation should be multiplied by a scaling factor. Given the introduction of the discretization step, we define T as the set of all time instants, from the very first time instant zero until the last possible time instant T_{Max}. Also, for every request $r \in R$, we define T^r as the set of all time instants in $[e^r, l^r - 1]$. For each request $r \in R$, the set T^r defines the time instants at which r could start travelling an arc without violating its latest time instant.

For this model, we use a space-time network, which means that every node is duplicated for each considered time instant. Also, for every arc $(i, j) \in A$, i.e. the physical network, there exist many arcs in the space-time network. In general, there exist one for each time instant. Arcs are modelled as follows: in the space-time network, each node $i \in N$ at time $t \in T$ is connected to each node $j \in N$ at time $t_2 \in T$ such that t_2 is equal to t plus the travel time from i to j at time t, i.e. δ_{ijt}. Each node $i \in N$ at time t is also connected to node $i \in N$ at time $t + 1$. These last arcs are used to model parking or passengers waiting for a(nother) vehicle. Fig. 5 displays a simple example of how to transform a standard network into a space-time network.

The structure of the space-time network allows to directly consider arcs with travel times depending on the departure time. We denote the set of all arcs in the space-time network by A^*. Since the travel times are embedded in the space-time formulation, routing variables embed timing causality. In fact, we introduce, both as routing and timing variables, binary variables x^r_{ijt} and y^v_{ijt}. The first variable assumes value one if request $r \in R$ travels arc $(i,j,t) \in A^*$ and zero otherwise; identically, the second variable assumes value one if vehicle $v \in V$ travels arc $(i,j,t) \in A^*$ and zero otherwise. In addition, we introduce binary variables a_{rv} and a_{rvt}. These variables are used to model request-vehicle pairing. In fact, a_{rvt} assumes value one if request $r \in R$ is carried by vehicle $v \in V$ at time $t \in T$ and a_{rv} assumes value one if request $r \in R$ was carried by vehicle $v \in V$ at any time $t \in T$. Finally, we introduce variable u^r for request $r \in R$ which assumes value one if the request is unserved, zero otherwise. These sets, parameters and variables are summarized in Table 3 and Table 4.

Although they may appear similar, our formulation strongly differs from time-index formulations (van den Bergh et al. (2016)) because, in our formulation, vehicles are allowed to travel the same arc multiple times. Moreover, some authors (Cortés et al. (2010); Masson et al. (2014)) modelled transfer nodes as a couple of dummy nodes (one node for the drop-off and one for the pick-up) connected by a direct arc with zero travel time. The main advantage of duplicating transfer nodes is that, in doing so, the chronological order within the transfer is respected by construction. In general, this is a useful property, but it is redundant in our case. In fact, in the discrete time case, the chronological order within the transfer is also respected by construction in the space-time network. In the continuous case instead, it is enforced due to explicit timing variables and constraints (Section 3.2.4). Introducing duplicates for the transfer nodes would not exclude the need for timing variables and constraints since we also aim at minimizing travel times, waiting times at transfer nodes and premature/late arrival and departure, all of which still has to be explicitly modelled via the timing variables and constraints that we introduce.

4.1. Core model

Before introducing the mathematical formulation, we explain how we model waiting and parking. We model parking of vehicle $v \in V$ at node $i \in N$ at time $t \in T$, by moving from node $i \in N$ at time $t \in T$ to node $i \in N$ at time $t + 1 \in T$; hence, variable y^v_{iit} assumes value one. The same holds for requests when they are waiting.

We employ a similar approach to determine early or late arrival and departure. For example, to know at what time request $r \in R$ arrives at its

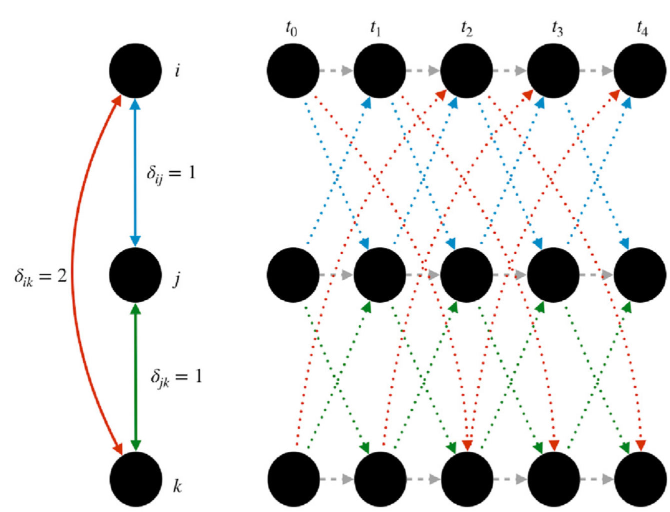


Fig. 5. Example of space-time network. On the left, the original network; on the right, its associated space-time network.

Table 3
Sets and parameters for the discrete time model.

T	set of all time instants.
T^r	set of time instants $t \in T$ such that $t \in [e^r, l^r - 1], \forall r \in R$.
A^*	set of arcs in the space-time network.
δ_{ijt}	travel time of arc $(i,j,t) \in A^*$.

Table 4
Variables for the discrete time model.

x^r_{ijt}	binary variable which is one if request $r \in R$ travels arc $(i,j,t) \in A^*$, zero otherwise.
y^v_{ijt}	binary variable which is one if vehicle $v \in V$ travels arc $(i,j,t) \in A^*$, zero otherwise.
a_{rvt}	binary variable which is one if vehicle $v \in V$ carries request $r \in R$ at time $t \in T$, zero otherwise.
a_{rv}	binary variable which is one if vehicle $v \in V$ carries request $r \in R$, zero otherwise.
u^r	binary variable which is one if request $r \in R$ is unserved, zero otherwise.

destination $d^r \in N$, we consider the values of the flows $x^r_{d^r d^r t}$ from $t = l^r$ backwards. The time instant $t \in T$ such that $x^r_{d^r d^r t-1} = 0$ and $x^r_{d^r d^r t} = 1$ is the time at which request $r \in R$ arrived at its destination $d^r \in N$. Additionally, if request $r \in R$ has preferred arrival time $pa^r \in T$ for destination $d^r \in N$, we compute $\sum_{t \geq pa^r}(1 - x^r_{d^r d^r t})$ to determine the number of time instants it was late. Similar ideas are used to determine early or late arrival, early or late departure, and also to establish waiting times at transfer nodes.

4.1.1. Objective function

The objective function minimizes the generalized cost Z (i.e. routing cost and costs related to a loss of quality of the service) and it is composed of eight terms. Each term penalizes one of the following: travel costs, passenger time spent in the vehicle, early or late arrival and departure times, number of transfers, passenger time spent waiting at transfer nodes, parking costs and unserved requests.

The first term determines the travel costs:

$$\alpha \sum_{v \in V} \sum_{(i,j,t) \in A^*} y^v_{ijt} l_{ij}.$$

The time spent by passengers in the vehicles is penalized in the second term:

$$\beta \sum_{r \in R} \sum_{(i,j,t) \in A^*, i \neq j} x^r_{ijt} \delta_{ijt} q^r.$$

The third and fourth terms are composed of two components each and determine the incurred penalty for departing and arriving early or late. To define the penalty for departing early, we use

$$\mu_1 \sum_{r \in R} \sum_{t < pd^r \in T^r} \left(1 - x^r_{o^r o^r t}\right) q^r$$

while the penalty if a client departs too late is regulated by

$$\mu_2 \sum_{r \in R} \sum_{t \geq pd^r \in T^r} x^r_{o^r o^r t} q^r.$$

The penalty for arriving early is given by

$$\mu_3 \sum_{r \in R} \sum_{t < pa^r \in T^r} x^r_{d^r d^r t} q^r$$

while the penalty for arriving late is defined by:

$$\mu_4 \sum_{r \in R} \sum_{t \geq pa^r \in T^r} \left(1 - x^r_{d^r d^r t}\right) q^r.$$

The fixed penalty for each transfer is given by the fourth term:

$$\eta \sum_{r \in R} \sum_{v \in V} a_{rv} q^r.$$

To consider the reduction of the quality of the service due to waiting at transfer nodes, we introduce

$$\gamma_1 \sum_{r \in R} \sum_{(i,i,t) \in A^*} x^r_{iit} q^r$$

with $i \neq o^r$ and $i \neq d^r$. To determine the parking cost, we add:

$$\gamma_2 \sum_{v \in V} \sum_{(i,i,t) \in A^*} y^v_{iit}.$$

Finally, to penalize the unserved requests we add:

$$E \sum_{r \in R} u^r q^r.$$

Hence, the generalized cost Z is given by:

$$\begin{aligned} Z = {} & \alpha \sum_{v \in V} \sum_{(i,j,t) \in A^*} y^v_{ijt} l_{ij} + \beta \sum_{r \in R} \sum_{(i,j,t) \in A^*, i \neq j} x^r_{ijt} \delta_{ijt} q^r + \mu_1 \sum_{r \in R} \sum_{t < pd^r \in T^r} \left(1 - x^r_{o^r o^r t}\right) q^r \\ & + \mu_2 \sum_{r \in R} \sum_{t \geq pd^r \in T^r} x^r_{o^r o^r t} q^r + \mu_3 \sum_{r \in R} \sum_{t < pa^r \in T^r} x^r_{d^r d^r t} q^r + \mu_4 \sum_{r \in R} \sum_{t \geq pa^r \in T^r} \left(1 - x^r_{d^r d^r t}\right) q^r \\ & + \eta \sum_{r \in R} \sum_{v \in V} a_{rv} q^r + \gamma_1 \sum_{r \in R} \sum_{(i,i,t) \in A^*} x^r_{iit} q^r + \gamma_2 \sum_{v \in V} \sum_{(i,i,t) \in A^*} y^v_{iit} + E \sum_{r \in R} u^r q^r. \end{aligned} \tag{59}$$

4.1.2. Routing and timing constraints

Because of the structure of the space-time network, routing constraints directly translate to timing constraints. For each request $r \in R$, we enforce its flow to start from its origin at $t = e^r$ (constraints (60) and (61)) and either to end at $t = l^r$ or to be unserved (constraints (62)).

$$\sum_{i,j \in N, i \neq o^r} x^r_{ije^r} = 0, \forall r \in R \tag{60}$$

$$\sum_{j \in N} x^r_{o^r je^r} = 1, \forall r \in R \tag{61}$$

$$\sum_{i \in N} x^r_{id^r t_2} + u^r = 1, \forall r \in R, t_2 | t_2 + \delta_{ijt_2} = l^r \tag{62}$$

For each vehicle, we impose that its route starts from its origin at time $t = 0$ (constraint (63) and (64)) and ends at $t = T_{Max}$ (constraint (65)).

$$\sum_{i,j \in N, i \neq o^v} y^v_{ij0} = 0, \forall v \in V \tag{63}$$

$$\sum_{j \in N} y^v_{o^v j0} = 1, \forall v \in V, \tag{64}$$

$$\sum_{i \in N} \sum_{j \in N} y^v_{ijt} = 1, \forall v \in V, \forall t \in T | t + \delta_{ijt} = T_{Max} \tag{65}$$

We establish flow conservation through constraints (66) and (67).

$$\sum_{i \in N} x^r_{ijt_2} = \sum_{i \in N} x^r_{jit}, \forall r \in R, \forall j \in N, \forall t \in T^r \tag{66}$$

$$\sum_{i \in N} y^v_{ijt_2} = \sum_{i \in N} y^v_{jit}, \forall v \in V, \forall j \in N, \forall t \in T / \{T^{max}\} \tag{67}$$

where $t_2 + \delta_{ijt_2} = t$ and $t_2 \in T$.

4.1.3. Pairing constraints

In this section, we explain how to pair vehicles and requests. Firstly, we have to impose that each request at any time instant can be paired to at most one vehicle (constraints (68)). A request can be paired with no vehicle for some time; for example, when it is waiting.

$$\sum_{v \in V} a_{rvt} \leq 1, \forall r \in R, \forall t \in T^r \tag{68}$$

Differently, a vehicle can be paired with more than one request, as long as this does not violate its capacity constraint, which is given by:

$$\sum_{r \in R} a_{rvt} q^r \leq q^v, \forall v \in V, \forall t \in T. \tag{69}$$

Also we have to impose that, as long as a vehicle and a request are paired, they have to travel the same route, i.e. if $a_{rvt} = 1$, then $x^r_{ijt} = y^v_{ijt}$, $\forall (i,j,t) \in A^*$. We do so by imposing

$$x^r_{ijt} \leq y^v_{ijt} + (1 - a_{rvt}), \forall r \in R, \forall v \in V, \forall (i,j,t) \in A^* \Big| t \in T^r \tag{70}$$

and

$$x^r_{ijt} \geq y^v_{ijt} - (1 - a_{rvt}), \forall r \in R, \forall v \in V, \forall (i,j,t) \in A^* \Big| t \in T^r. \tag{71}$$

Yet, constraints (70) and (71) can be equivalently rewritten as:

$$\sum_{j \in N} x^r_{ijt} \leq \sum_{j \in N} y^v_{ijt} + (1 - a_{rvt}), \forall r \in R, \forall v \in V, \forall i \in N, \forall t \in T^r, \tag{72}$$

$$\sum_{j \in N} x^r_{ijt} \geq \sum_{j \in N} y^v_{ijt} - (1 - a_{rvt}), \forall r \in R, \forall v \in V, \forall i \in N, \forall t \in T^r, \tag{73}$$

$$\sum_{i \in N} x^r_{ijt} \leq \sum_{i \in N} y^v_{ijt} + (1 - a_{rvt}), \forall r \in R, \forall v \in V, \forall j \in N, t \in T^r, \tag{74}$$

$$\sum_{i \in N} x^r_{ijt} \geq \sum_{i \in N} y^v_{ijt} - (1 - a_{rvt}), \forall r \in R, \forall v \in V, \forall j \in N, \forall t \in T^r. \tag{75}$$

Constraints (70)–(75) are always true when $a_{rvt} = 0$. The first formulation (70) and (71) is composed by $2 \cdot |N|^2 \cdot |T^r| \cdot |R| \cdot |V|$ constraints, while the second formulation (72)–(75) requires only $2/\ |N|$ of those constraints. Even though fewer constraints do not necessary mean that the problem is easier to solve (valid inequalities are an example), in this case it does.

Additionally, to enforce that whenever a request is not paired to any vehicle, it has to wait, we impose constraints (76).

$$\sum_{i \in N} x^r_{iit} \geq 1 - \sum_{v \in V} a_{rvt}, \forall r \in R, \forall t \in T^r \tag{76}$$

Finally, to model that each request may have a maximum of b^r transfers, we introduce constraints (77) and (78):

$$B a_{rv} \geq \sum_{t \in T} a_{rvt}, \forall v \in V, \forall r \in R, \tag{77}$$

$$\sum_{v \in V} a_{rv} - 1 \leq b^r, \forall r \in R. \tag{78}$$

4.2. Model extension

As for the continuous time case (Section 3.3), we show how to extend the model in order to consider people dependent service time (time for get-in operations). Differently with respect to the continuous time model, nonconstant travel times are directly embedded in the structure of the space-time network.

4.2.1. People dependent service times

In this section, we present how to introduce people dependent service times. We can set any finite integer number of time steps (s^r) as service times through constraints (79).

$$\sum_{i \in N} x^r_{iit} \geq a_{rvt} - a_{rvt-s^r}, \forall t \in T^r | t - s^r \in T^r, \forall r \in R, \forall v \in V \tag{79}$$

If, at the same time, more than one request are picked up, only the longest one is considered. How to model the sum of all service times instead of the longest service time is shown in Appendix A.2. Fig. 6 shows an example of how constraints (79) work. In this example, there are three requests; request r_0 is already in the taxi at time t_0 and it dropped off at time t_1. The other two requests, r_1 and r_2, can be picked up at t_1 the earliest (the time the taxi arrives in $i \in N$). Nevertheless, the service time of r_2 ($s^{r_2} = 4$) constraints the vehicles to move at t_5 the earliest. Hence, while the pick up process of r_2 starts immediately, the boarding process of r_1 (service time $s^{r_1} = 2$) starts at time t_3.

Also, introducing service times leads to the following modifications of the two terms related to departure in the objective function (59):

$$\sum_{r \in R} \sum_{t < (pd^r + s^r) | t \in T^r} \mu_1 (1 - x^r_{o^r o^r t}) q^r$$

and

$$\sum_{r \in R} \sum_{t > (pd^r + s^r) | t \in T^r} \mu_2 x^r_{o^r o^r t} q^r.$$

In fact, the service time causes the departure time and the time at which the node is left to differ by s^r time instants. Given the different approach in modeling early and late departure, these modifications are not needed in the continuous time model.

5. Computational experiments

Even though the DARPT is an NP-hard problem, the number of variables and constraints in both the continuous time and discrete time formulations are bounded by polynomial functions in the size of the problem. Table 5 summarizes variables in the continuous and in the discrete model.

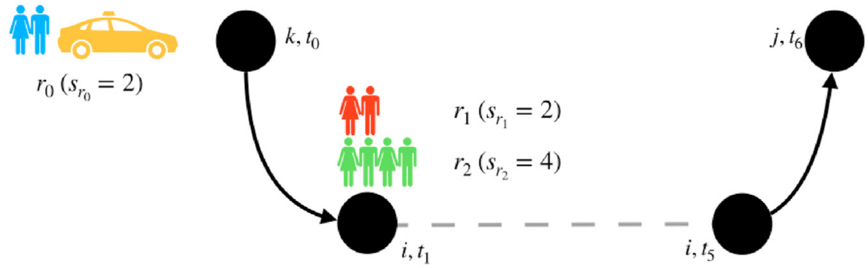
In order to compare the models, we assume the discrete time step to be ε; then, it holds that $|M^v| = |T|$ and $|M^r| = |T^r|$. It follows that the variables in the continuous time model are more than the ones in the discrete time model. In fact, the continuous time model is composed of $3 \cdot |R| \cdot |T^r| + 3 \cdot |V| \cdot |T| + |R| \cdot |A| \cdot |T^r| + |V| \cdot |A| \cdot |T| + 5 \cdot |R| + |R| \cdot |V| + |R| \cdot |T^r| \cdot |A| \cdot |K| + |V| \cdot |T| \cdot |A| \cdot |K| + |R| \cdot |T^r| \cdot |V| \cdot |T|$ variables (of which $2|R| \cdot |T^r| + 2|V| \cdot |T| + 4|R|$ are continuous), while the discrete time model has $|R| \cdot |A| \cdot |T^r| + |V| \cdot |A| \cdot |T| + |R| \cdot |V| \cdot |T^r| + |R|$ variables (all binary). Although these are two (almost) equivalent models, the continuous time one has $2 \cdot |R| \cdot |T^r| + 2 \cdot |V| \cdot |T| + 4 \cdot |R|$ continuous variables more and $|R| \cdot |V| \cdot |T|(|T^r|-1) + |V| \cdot |T^v| + |R| \cdot |T^r| + |V| \cdot |T^v| \cdot |A| \cdot |K| + |R| \cdot |T^r| \cdot |A| \cdot |K|$ binary variables more (Table 5).

Table 6 shows the number of constraints, divided by type, in both models. Also in this case, the discrete time model has less constraints with respect to its continuous counterpart.

5.1. Benchmark

To test the models, we apply the minor changes explained in Appendix C; then, we create the following benchmark, based on real-life data. In general, in order to be useful, transfers ask the length of the trips to be longer than the deviation and the stop of the vehicle. Hence, we choose to test our models with interurban trips. We select the twenty most populated cities (the central station of each city has been considered as the exact coordinate) in the most densely populated province of the Netherlands, i.e. South Holland. In particular, we choose: Rotterdam, Delft, Capelle aan den IJssel, Schiedam, Gouda, The Hague, Rijswijk, Voorburg, Leiden, Zoetermeer, Dordrecht, Zwijndrecht, Gorinchem, Spijkenisse, Vlaardingen, Barendrecht, Maassluis, Alphen aan den Rijn, Ridderkerk and Papendrecht. We connect these cities as shown in Fig. 7.

We collected travel data, i.e. length and time of each arc, from Google Maps. On July 17, 2019 we acquired (expected) travel times for July 18, 2019 from 8:00AM to 9:45AM with time intervals of 15 min each. The chosen day is an average commuting day (Thursday) where no major



	t_0	t_1	t_2	t_3	t_4	t_5	t_6
a_{r_0vt}	1	0	0	0	0	0	0
$a_{r_0vt} - a_{r_0vt-s_{r_0}}$	0	0	-1	0	0	0	0
a_{r_1vt}	0	0	0	1	1	1	1
$a_{r_1vt} - a_{r_1vt-s_{r_1}}$	0	0	0	1	1	0	0
a_{r_2vt}	0	1	1	1	1	1	1
$a_{r_2vt} - a_{r_2vt-s_{r_2}}$	0	1	1	1	1	0	0

Fig. 6. Example of service times. Request r_0 is already on vehicle v at time t_0 and drops off at time t_1. Request r_1 has a service time $s^{r_1} = 2$ time instant and leaves node i at time t_5. Request r_2 has a service time $s^{r_2} = 4$ time instant and leaves node i at time t_4.

Table 5
Variables in continuous and discrete time model.

Continuous time model	Discrete time model
t_m^r	–
t_m^v	–
w_m^r	–
w_m^v	–
x_{ijm}^r	x_{ijt}^r
y_{ijm}^v	y_{ijt}^v
c_r^+	–
c_r^-	–
d_r^+	–
d_r^-	–
a_{rv}	a_{rv}
$p_{rm_rvm_v}$	a_{rvt}
$g_{m_v}^v$	–
$g_{m_r}^r$	–
TT_{ijmk}^v	–
TT_{ijmk}^r	–
u^r	u^r

Table 6
Constraints in continuous and discrete time model.

Type of constraints	Continuous time model	Discrete time model
Routing	$2\|R\| \cdot \|T^r\| + 2\|R\| + 2\|V\| \cdot \|T\| + \|V\|$	$3\|R\| + \|R\| \cdot \|N\| \cdot \|T^r\| + 3\|V\| + \|V\| \cdot \|N\| \cdot \|T\|$
Timing	$4\|R\| \cdot \|T^r\| + 2\|R\| + 4\|V\| \cdot \|T\| + 2\|V\|$	–
Departure and arrival	$8\|R\|$	–
Pairing	$2\|R\| \cdot \|T^r\| + \|V\| \cdot \|T\| + \|R\| \cdot \|V\| + 2\|A\| \cdot \|R\| \cdot \|T^r\| \cdot \|V\| \cdot \|T\| + 2 \cdot \|R\| \cdot \|T^r\| \cdot \|V\| \cdot \|T\| + \|R\|$	$2\|R\| \cdot \|T^r\| + \|V\| \cdot \|T\| + 4\|R\| \cdot \|V\| \cdot \|N\| \cdot \|T^r\| + \|R\| \cdot \|V\| + \|R\|$
People depending service time	$\|R\| + \|R\| \cdot \|T^r\| + \|V\| \cdot \|T\| + \|R\| \cdot \|T^r\| \cdot \|V\| + 3\|R\| \cdot \|V\| \cdot \|T\| + \|R\| \cdot \|T^r\| \cdot \|V\| \cdot \|T\|$	$\|R\| \cdot \|V\| \cdot \|T^r\|$
Nonconstant travel time	$2\|R\| \cdot \|T^r\| \cdot \|V\| \cdot \|T\| + \|R\| \cdot \|T^r\| \cdot \|A\| + 2\|R\| \cdot \|T^r\| \cdot \|TS\| + 2\|V\| \cdot \|T\| \cdot \|TS\| + \|V\| \cdot \|T\| \cdot \|A\| + \|R\| \cdot \|T^r\| + \|V\| \cdot \|T\|$	–

road blocks were present. In our tests, we used a time step of 1 min; hence, we set travel times for the time instants in between two collected data points to the value of the earliest data point. On this map, we test ten times each of the following cases: V2R3, V2R4, V2R5, V3R4, V3R5, V3R6, V3R7, V4R5, V4R6, V4R7, V4R8, V4R9. By V*i*R*j*, we mean that there are *i* vehicles and *j* requests in the instance.

Vehicles are initially positioned to give a good coverage of the area. In particular, we position the first vehicle in Dordrecht, the second one in Vlaardingen, the third one (if any) in Alphen aan den Rijn and the last one (if any) in Rotterdam. The positions of the vehicles are indicated by the black nodes in Fig. 8. The capacity q^v of each vehicle is set to 6.

We set the destination d^r of each request to The Hague (the green node in Fig. 8) and the latest arrival time l^r to 10:00. This increases the chances that the last parts of some requests' routes overlap in space and are close in time. This makes transfers more likely to happen.

All requests have randomly chosen parameters such that the origin o^r is different from The Hague and such that the time preferences are coherent. In fact, we set the earliest time e^r to be earlier than the preferred departure time pd^r which in turn should be earlier than the preferred arrival time pa^r which also has to be earlier than the latest arrival time l^r. In addition, we set the preferred arrival time pa^r to be later than the preferred departure time pd^r plus the service time s_r. In formulas, this translates to: $e^r \leq pd^r \leq pd^r + s_r \leq pa^r \leq l^r$. These random choices are repeated until $1.2SPP \leq l^r - e^r \leq 1.5SPP$, where SPP is the shortest possible time to go from origin o^r to destination d^r (in the constant travel time configuration). For each request, the maximum number of transfers b^r and the party size q^r are uniformly randomly chosen among {1,2,3}. Service times s_r are set to q^r minutes.

All the above described instances and their logs with detailed solutions are available at the following website (http://doi.org/10.4121/uuid:1ad27269-cdcf-43ed-a639-8fea09c48449). The instances are tested in continuous and discrete time, both in the core and enlarged configurations. For comparison reasons, we also test the same instances in a no transfer setting by fixing the maximum number of transfers b^r to zero.

5.2. Tuning parameters

To tune the parameters, we refer mainly to Correia and van Arem (2016) which, in turn, estimated the cost parameters based on a semi-real case study of private and public transportation in the Netherlands. The main differences and additions are the values of the travel costs α, the

Fig. 7. South Holland region with our network.

(a) Two unrelated transfers (b) Two sequential transfers

Fig. 8. Examples of transfers. The colored arcs indicate the path of each vehicle and their origin nodes are denoted by the black nodes. The origin nodes of the requests are depicted by yellow nodes while the green node indicates their final destination. The red nodes represent nodes where a transfer happens and the grey nodes represent nodes that are simply passed by the vehicles. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

cost of a transfer η and the cost of waiting at a transfer node γ_1. The value of the travel cost α was set to €1 per *km*, instead of €0.1 per *km*, because this is closer to the prices for Dutch taxis (note that Correia and van Arem (2016) referred to privately owned vehicles, not an on-demand service). Also, we set the cost of waiting at a transfer node γ_1 equal to €1.106 per minute, in between the cost of the time spent inside a vehicle $\beta =$ €0.806 per minute and the cost of premature departure $\mu_1 =$ €1.306 per minute. In addition, we set the cost of late arrival μ_4 to the cost of premature departure μ_1 and the costs of late departure μ_2 and premature arrival μ_3 to €0.306 per minute. Finally, the cost of each transfer η is set to €1 and the penalty E for each unserved request to €999. Table 7 displays all the values of the parameters.

5.3. Tests

All tests are run on a Linux machine with the following architecture: x86_64, 4 CPUs (Intel(R) Core(TM) i5 CPU 660 @ 3.33 GHz). It ran code written in Python 2.7 through the Spyder interface and adopting Gurobi

7.5.2 as the MILP solver. A time limit of 1 h was set for the solution of each MILP.

5.3.1. Examples

Fig. 8 shows the results for the third and eighth instance of $V4R7$. In both cases, 4 vehicles are used to serve 5 requests (two requests are unserved). The grey nodes represent nodes that are simply passed by the vehicles (no pick up or transfers); the colored arcs indicate the path of each vehicle (one colour per vehicle) while their origin nodes are denoted by the black nodes. The red nodes represent nodes where a transfer happens, the origin nodes of the requests are depicted by yellow nodes while the green node indicates their final destination. The three nodes with double colors represent nodes where multiple actions happen. In particular in Fig. 8a, the yellow and red node denotes the origin node of a request as well as a transfer node; in Fig. 8b, the yellow and black node denotes the origin node of a request as well as the origin node of a vehicle while the red and black node describes the origin node of a vehicle as well as a transfer node.

Fig. 8a shows a transfer between the green and the pink vehicle and a transfer between the yellow and the blue vehicle. Fig. 8b shows a sequence of two transfers, from the green and the pink vehicle to the blue vehicle.

5.3.2. Average results

Table 8 shows the average results of the conducted tests. We remind that, for each scenario, 10 different instances were generated. The rows illustrate the features of the solutions. The first row indicates the number of vehicles $|V|$ and requests $|R|$ in the instances. The following eight rows specify how many instances were solved to optimality within the time limit for the various cases. The cases differ in continuous (C) and discrete (D) time model, core (Core) and enlarged (Enl) model, and transfers (T) and no transfers (No-T) case.

Then, the following four rows indicate how many instances present at least one transfer. The number in brackets indicates how many of the instances not solved to optimality present at least one transfer in their incumbent solution. After showing how many instances have transfers, we display the total number of transfers happening over all instances per case. Similarly as before, the number in brackets indicate how many transfers happen in the incumbent solution of the instances not solved to optimality.

Succeeding the rows on transfers, we present features regarding the computation time and the objective function for the continuous time and discrete time models where transfers are allowed. The presented values of the objective functions and computation times are the averages over the values of the instances that both models could solve to optimality. In some cases, one of the two models could not solve any instance. Only in these cases, the average objective values and computation times of the model that could solve some instances are calculated on all these instances solved to optimality.

The following four rows indicate the gap percentage in the objective function and computation time between the continuous time and discrete time models. Each parameter is computed as $\frac{V_D - V_C}{V_D}$, where V_D can assume either the value of the objective function or the computational time of the discrete time model and V_C assumes either the value of the objective function or the computational time of the continuous time model.

Finally, the last eight rows assess the benefits (in the objective functions) of introducing transfers. Since setting $b^r = 0$ is not the best way to model the standard DARP without transfers, we do not present the computational time of the no transfers case. In these rows, each parameter is computed as $\frac{V_{No-T} - V_T}{V_T}$, where V_T assumes the value of the objective function in the models where transfers are allowed while V_{No-T} assumes the value of the objective function of the models where transfers are not allowed.

For all models, we notice that when an empty vehicle has to move, it tries to do so during a traffic jam. This can easily be explained by looking at the objective function. Since for empty vehicles travelling costs are only related to travel distance and not to travel time, vehicles try to maximize their commuting time to minimize their parking fees. This does not hold when a request is being served by the vehicle, because a time-related penalty has to be paid.

Continuous time instances may achieve lower objective function values because of the quality loss implied in discretization itself. For example, if the best departure time for a request is generic t, in discrete time, it would need to rely on t's closest time instant. A time step of 1 min is small enough to curtail the error embedded in the discretization itself. In fact, the gap in the objective function is at most 0.39%.

Analyzing the results, we can state that, in general, discretizing the time results in slightly worse solutions in terms of objective function value but better results in terms of computational times. Although most of the discretized instances were solved in considerably less time than their continuous counterpart, this seems not to hold when the size of the problems increases. Indeed, the continuous time model solved more instances to optimality (and often in less time) in every scenario where 4 vehicles are present. This confirms the conclusions of van den Berg and van Essen (2019) which states that for different conditions, the continuous or discrete model could yield better results in terms of computation time.

Since more instances are solved within the time limit in the core models compared to the enlarged models, we can deduce that including service times and variable travel times increases the level of difficulty in solving the problem.

Although transfers are thought to save travel costs, this does not always show in randomly created small instances. Nevertheless, transfers improve the average best solution by 12.24% in $V3R7$ (for the enlarged discrete case). In this case, one instance can serve all requests when transfers are allowed while it can serve all requests but one when transfers are not allowed. This creates the difference in the objective function values. It may seem that, in some cases, allowing transfers makes the problem easier (for instance, in continuous enlarged $V4R6$, more instances were solved when allowing transfers). Since the models and instances used (both for the transfer and the no transfers case) are designed to include and favour transfers, this comparison would be unfair.

The number of transfers (considering also transfers in the incumbent solutions) seems to increase steadily as the instance size grows. This is a very promising feature of this problem. We conjecture that transfers can lead to considerable savings in bigger instances.

Table 7
Values of the parameters.

Name	Value	Description
A	1 € per *km*	Travel cost per kilometer.
B	0.806 € per *min*	Cost of time spent inside a vehicle per minute per person.
γ_1	1.106 € per *min*	Cost of waiting at transfer nodes per minute per person.
γ_2	1.81 € per *hour*	Cost of parking per minute.
μ_1	1.306 € per *min*	Cost of premature departure per minute per person.
μ_2	0.306 € per *min*	Cost of late departure per minute per person.
μ_3	0.306 € per *min*	Cost of premature arrival per minute per person.
μ_4	1.306 € per *min*	Cost of late arrival per minute per person.
H	1 €	Cost of transfer.
E	999 €	Penalty per person per unserved request.

6. Conclusions

In this paper, we described and tested two mixed integer linear models, and their extensions, for the DARPT where cycles are allowed. Additionally, we introduced the ‘move’ concept. This is useful when modelling loops and relating continuous variables (timing variables) to binary ones (routing and causality variables). We showed how much transfers increase the complexity of the problem (more instances were

Table 8
Computational results.

size	V2R3	V2R4	V2R5	V3R4	V3R5	V3R6	V3R7	V4R5	V4R6	V4R7	V4R8	V4R9
# Instances Opt C T Core	9/10	7/10	6/10	10/10	8/10	8/10	3/10	9/10	9/10	6/10	3/10	2/10
# Instances Opt D T Core	10/10	10/10	10/10	10/10	10/10	9/10	9/10	5/10	5/10	5/10	2/10	0/10
# Instances Opt C T Enl	9/10	3/10	4/10	8/10	3/10	1/10	1/10	7/10	5/10	4/10	1/10	1/10
# Instances Opt D T Enl	10/10	10/10	10/10	10/10	10/10	8/10	10/10	2/10	1/10	0/10	0/10	0/10
# Instances Opt C No-T Core	10/10	8/10	7/10	10/10	9/10	8/10	3/10	9/10	7/10	2/10	2/10	0/10
# Instances Opt D No-T Core	10/10	10/10	10/10	10/10	10/10	10/10	10/10	8/10	8/10	7/10	6/10	4/10
# Instances Opt C No-T Enl	8/10	5/10	3/10	8/10	5/10	0/10	1/10	2/10	1/10	0/10	0/10	0/10
# Instances Opt D No-T Enl	10/10	10/10	10/10	10/10	10/10	10/10	10/10	7/10	8/10	3/10	1/10	2/10
# Instances Trans C T Core	1 (0)	0 (1)	1 (0)	1 (0)	2 (0)	1 (0)	0 (0)	4 (0)	3 (1)	3 (4)	2 (3)	1 (6)
# Instances Trans D T Core	1 (0)	1 (0)	1 (0)	1 (0)	2 (0)	1 (0)	1 (0)	2 (2)	2 (2)	4 (2)	1 (4)	0 (4)
# Instances Trans C T Enl	0 (0)	0 (1)	2 (0)	0 (0)	0 (1)	1 (0)	0 (2)	3 (0)	2 (2)	3 (6)	1 (3)	0 (6)
# Instances Trans D T Enl	0 (0)	0 (0)	1 (0)	0 (0)	1 (0)	1 (0)	1 (0)	0 (3)	0 (4)	0 (5)	0 (4)	0 (4)
# Trans C T Core	1 (0)	0 (2)	1 (0)	1 (0)	2 (0)	1 (0)	0 (0)	4 (0)	3 (1)	3 (6)	2 (3)	1 (8)
# Trans D T Core	1 (0)	2 (0)	1 (0)	1 (0)	2 (0)	1 (0)	1 (0)	2 (2)	2 (2)	6 (2)	1 (8)	0 (4)
# Trans C T Enl	0 (0)	0 (2)	2 (0)	0 (0)	0 (1)	1 (0)	0 (2)	3 (0)	2 (2)	3 (8)	1 (3)	0 (8)
# Trans D T Enl	0 (0)	0 (0)	1 (0)	0 (0)	1 (0)	1 (0)	1 (0)	0 (3)	0 (4)	0 (7)	0 (8)	0 (7)
Time C T Core	112.92	1442.01	988.76	490.19	1001.09	1036.39	466.98	1265.61	967.57	2691.73	1425.74	2905.04
Time D T Core	9.99	41.67	413.77	231.87	186.61	826.32	190.60	1733.12	1446.40	2646.50	1739.84	–
Time C T Enl	787.35	481.26	1346.05	1888.43	1300.43	3348.60	2329.74	2671.78	1890.01	3254.65	2213.50	3187.35
Time D T Enl	13.80	41.69	202.08	339.98	223.91	335.19	507.51	121.36	2450.13	–	–	–
Obj C T Core	640.67	1227.71	2495.72	593.31	830.24	1081.63	760.88	819.11	102.77	1157.01	1032.88	1312.12
Obj D T Core	640.67	1227.71	2495.72	593.31	830.24	1081.63	760.88	819.11	104.06	1159.16	1036.92	–
Obj C T Enl	868.60	519.04	2418.89	582.16	679.04	958.48	861.21	455.01	1073.34	1219.85	1259.39	1353.13
Obj D T Enl	871.19	521.86	2420.95	582.63	679.13	960.59	861.46	455.51	1075.59	–	–	–
Gap C/D T Obj Core (%)	0	0	0	0	0	0	0	0	0.13	0.19	0.39	–
Gap C/D T Time Core (%)	−1030.74	−3360.56	−138.97	−111.41	−436.46	−25.42	−145.00	26.98	33.10	−1.71	18.05	–
Gap C/D T Obj Enl (%)	0.30	0.54	0.09	0.08	0.01	0.22	0.03	0.11	0.21	–	–	–
Gap C/D T Time Enl (%)	−5605.46	−1054.25	−566.10	−455.46	−480.77	−899.00	−359.06	−2101.47	22.86	–	–	–
Gap T/No-T C Obj Core (%)	0.06	0.00	2.22	0.04	0.08	0.04	0.00	0.44	0.25	0.11	0.02	–
Gap T/No-T C Obj Enl (%)	0.00	0.00	0.02	0.00	0.00	–	0.00	0.00	0.00	–	–	–
Gap T/No-T D Obj Core (%)	0.04	0.01	1.40	0.04	0.06	0.03	6.31	0.54	0.12	0.26	0.23	–
Gap T/No-T D Obj Enl (%)	0.00	0.00	1.21	0.00	11.23	0.02	12.24	0.00	0.00	–	–	–

solved to optimality in the no transfers case compared to the case where transfers were allowed) and how much cost savings (in the objective functions) they can lead to. Also, we illustrated a method to create instances based on Google Maps data. Clearly, this method can easily scale up and create real-life sized instances which would be useful in practice for testing (meta)heuristics.

All the models proposed in this paper allow for a great deal of flexibility. In fact, requests and vehicles can have different dimensions and, in general, there is no need for parameters $\alpha, \beta, \gamma_i, \mu_i$ to be constant or equal for every passenger or vehicle. In practical applications, it would be possible to dynamically tune these parameters depending on the history of a customer. In such a way, it is possible to ensure an equally distributed quality of service among clients.

Interestingly, transfers are not explicitly modelled. In fact, transfers and their related complexity are embedded in the *flow formulation* of the vehicles and of the requests. This formulation easily allows for sequences of transfers and transfers with multiple vehicles. Also, with this *flow formulation*, the number of variables is not affected by how many transfers are allowed. However, since the resulting models have many equality constraints, they may be computationally more challenging than formulations where only inequality constraints are used.

Some of the small instances were not solved to optimality within the given time limit. This was expected since we faced a more complex variant of an infamous NP-hard problem. Nevertheless, the sizes of the solved instances are in the range of the ones solved in the literature for similar problems.

This paper lays the foundations for other interesting developments such as the offline large-scale DARPT and its online counterpart. Most likely, these problems can be solved only with heuristic methods.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

A. Modifications about people dependent service times

In Sections 3.3.1 and 4.2.1, we explained how to include people dependent service times under the assumption that, if multiple requests are picked up simultaneously, only the largest service time is considered. In this appendix, we consider the same problem under the assumption that, if multiple requests are picked up simultaneously, their sum is considered.

A.1 Continuous time model

The constraints related to the requests ((38)–(41)) hold also in this case while the ones related to the vehicles ((42)–(47)) are modified as follows.

To ensure that h_{vm_vr} assumes value one whenever request $r \in R$ is picked up by vehicle $v \in V$ at move $m_v \in M^v$, we impose the following three constraints: constraints (80) for the first move of a vehicle, constraints (81) for the first move of a request and constraints (82) for all other moves of requests and vehicles.

$$h_{v1r} \geq \sum_{m_r \in M^r} p_{rm_rv1}, \forall v \in V, \forall r \in R \tag{80}$$

$$h_{vm_vr} \geq p_{r1vm_v}, \forall v \in V, \forall m_v \in M^v, \forall r \in R \tag{81}$$

$$h_{vm_vr} \geq \left(p_{rm_rvm_v} - p_{rm_r-1vm_v-1}\right), \forall v \in V, \forall m_v \in M^v \setminus \{1\}, \forall r \in R, \forall m_r \in M^r \setminus \{1\} \tag{82}$$

To ensure a tight upper bound, we impose constraint (83). Since $\sum_{v \in V}\sum_{r \in R} a_{vr}s_r$ is exactly the sum of all service times that actually took place, constraint (83) prohibits h_{vmr} to assume value one just to have conveniently large waiting times to better fit time preferences.

$$\sum_{v \in V}\sum_{m_v \in M^v}\sum_{r \in R} h_{vm_vr}s_r \leq \sum_{v \in V}\sum_{r \in R} a_{vr}s_r \tag{83}$$

The modified timing constraints (48) remain unchanged.

A.2 Discrete time model

For the discrete time model, we introduce binary variable h_{rvt} which assumes value one if request $r \in R$ is being picked up by vehicle $v \in V$ at time $t \in T$. Hence, we substitute constraints (79) by:

$$h_{rvt} \geq a_{rvt} - a_{rvt-s_r}, \forall r \in R, \forall v \in V, \forall t \in T^r | t - s_r \in T^r \tag{84}$$

and

$$\sum_{r \in R} h_{rvt} \leq 1, \forall v \in V, \forall t \in T. \tag{85}$$

This allows to consider one request at a time, hence the total service time experienced becomes the sum of the single service times. To ensure a tight upper bound, we impose constraint (86).

$$\sum_{v \in V}\sum_{t \in T^r}\sum_{r \in R} h_{rvt}s_r \leq \sum_{v \in V}\sum_{r \in R} a_{vr}s_r \tag{86}$$

B. Idling, parking, transit and transfer nodes

In this appendix, we define constraints to model the nodes with restricted accessibility. This is useful, for instance, to model nodes situated near historical attractions. Requests can easily be directed there but parking is not allowed. For the sake of ease, we define:

– an idling node as a node where a request can wait but a vehicle cannot park.
– a parking node as a node where a request cannot wait but a vehicle can park.
– a transit node as a node where a request cannot wait and a vehicle cannot park.
– a transfer node as a node where a request can wait and a vehicle can park.

B.1 Continuous time model

To model this in the continuous time model, we impose:

– $w^v = 0$ for every idling and transit node.
– $w^r = 0$ for every parking and transit node.

Given a disjoint partition N_1 of idling nodes, N_2 of parking nodes, N_3 of transit nodes and N_4 of transfer nodes, we impose the following set of constraints:

$$w_m^r \leq \sum_{(i,j) \in A | j \in N_1 \cup N_4} x_{ijm}^r B, \forall r \in R, m \in \{0\} \cup M^r. \tag{87}$$

We impose a similar set of constraints for parking, with the addition that a vehicle should still be able to stop to pick a request up at any idling node.

$$w_m^v \leq \sum_{(i,j)\in A|j\in N_2\cup N_4} y_{ijm}^v B + \sum_{r\in R} h_{vmr} s_r, \forall v\in V, \forall m\in \{0\}\cup M^v \quad (88)$$

B.2 Discrete time model

To model idling, parking, transit and transfer nodes in the discrete time model, we impose:

- $y_{iit}^v = 0$ for every idling and transit node.
- $x_{iit}^r = 0$ for every parking and transit node.

Given a disjoint partition $N_1$of idling nodes, oN_2f parking nodes, ofN_3 transit nodes and of N_4transfer nodes, we impose the following sets of constraints:

$$\sum_{r\in R} \sum_{(i,i,t)\in A^*|i\in N_2\cup N_3, t\in T^r} x_{iit}^r = 0 \quad (89)$$

and

$$\sum_{t\in T} y_{iit}^v \leq \sum_{r\in R} h_{rvt}, \forall i\in N_1\cup N_3, \forall v\in V, \forall t\in T. \quad (90)$$

C. Equivalent models

Even though the continuous and discrete time model share the same foundations, sometimes their objective functions differ despite having the same routing and timing solution. This appendix explains how to modify the models such that, for the same solution, they return the same objective function value. These modifications do not affect the values of the routing and timing variables, which are already equivalent for the two models.

C.1 Unserved requests and late arrival

When a request is not served, a penalty has to be paid. In the continuous time model, this is in addition to the (presumed) late departure and arrival term. Since the penalty for late departure is irrelevant with respect to the one for unserved requests, this does not affect the general routing. In order to obtain from both models the same objective function value, we modify constraints (26) into:

$$c_r^+ \geq t_{\mathscr{M}_v}^r - Bu_r, \forall r\in R. \quad (91)$$

Similarly, we modify constraints (24) into:

$$d_r^- \leq t_0^r + w_0^r + Bu_r, \forall r\in R. \quad (92)$$

C.2 Objective functions

Only in the discrete time model, the service time is considered as waiting time; therefore, vehicles have to pay parking costs during the service times. To balance this difference, we add

$$+\gamma_2 \sum_{r\in R}\sum_{v\in V}\sum_{m\in M^v} h_{rvm} s_r$$

to the objective function of the continuous time model, where γ_2 represents the parking costs per minute.

Finally, we add the following term to the continuous time objective function:

$$+\gamma_1 \sum_{r\in R}\sum_{v\in V}\sum_{m\in M^v} h_{rvm} s_r.$$

By doing so, the service time of a request is equally penalized by a γ_1 factor in both models when the request is waiting at a transfer node. Recall that γ_1 represents the cost of waiting at a transfer node per minute per person.

D. Complete models

In this appendix, we present the complete models. This appendix does not add or modify any constraints with respect to the ones previously introduced.

D.1 Continuous time model - Core

$$\begin{aligned}\text{minimize} \quad Z = {} & \alpha\sum_{v\in V}\sum_{(i,j)\in A}\sum_{m\in M^v} y^v_{ijm} l_{ij} + \beta\sum_{r\in R} q^r\left(t^r_{\mathscr{M}_r} - \sum_{m\in\{0\}\cup M^r} w^r_m - t^r_0\right)\\ & +\mu_1\sum_{r\in R}\left(pd^r - d^-_r\right) + \mu_2\sum_{r\in R}\left(d^+_r - pd^r\right) + \mu_3\sum_{r\in R}\left(c^+_r - pa^r\right)\\ & +\mu_4\sum_{r\in R}\left(pa^r - c^-_r\right) + \eta\sum_{r\in R}\sum_{v\in V} a_{rv} q^r + \gamma_1\sum_{r\in R}\sum_{m\in M^r} q^r w^r_m\\ & +\gamma_2\sum_{v\in V}\sum_{m\in\{0\}\cup M^v} w^v_m + E\sum_{r\in R} u^r q^r \end{aligned} \tag{1}$$

such that

$$\sum_{(i,j)\in A} x^r_{ijm} \le 1 \quad \forall r\in R, \forall m\in M^r \tag{2}$$

$$\sum_{(o^r,j)\in A} x^r_{o^r j1} \ge \sum_{(i,j)\in A} x^r_{ij1} \quad \forall r\in R \tag{3}$$

$$\sum_{(i,j)\in A} x^r_{ijm} = \sum_{(j,k)\in A} x^r_{jk(m+1)} \quad \forall r\in R, \forall m\in M^r\setminus\{\mathscr{M}^r\}, j\ne d^r \tag{4}$$

$$u^r + \sum_{m\in M^r}\sum_{(i,d^r)\in A} x^r_{id^r m} = 1 \quad \forall r\in R \tag{5}$$

$$\sum_{(i,j)\in A} y^v_{ijm} \le 1 \ \forall v\in V, \ \forall m\in M^v \tag{6}$$

$$\sum_{(o^v,j)\in A} y^v_{o^v j1} \ge \sum_{(i,j)\in A} y^v_{ij1} \quad \forall v\in V \tag{7}$$

$$\sum_{(i,j)\in A} y^v_{ijm} \ge \sum_{(j,k)\in A} y^v_{jk(m+1)} \ \forall v\in V, \forall m\in M^v\setminus\{\mathscr{M}_v\}, \ \forall j\in N \tag{8}$$

$$t^r_m \ge 0 \ \forall r\in R, \ \forall m\in\{0\}\cup M^r \tag{9}$$

$$w^r_m \ge 0 \ \forall r\in R, \ \forall m\in\{0\}\cup M^r \tag{10}$$

$$t^r_{m+1} = t^r_m + w^r_m + \sum_{(i,j)\in A} x^r_{ijm}\delta_{ij} \quad \forall r\in R, \forall m\in\{0\}\cup M^r\setminus\{\mathscr{M}_r\} \tag{11}$$

$$t^r_0 = e^r \quad \forall r\in R \tag{12}$$

$$t^r_{\mathscr{M}_r} \le l^r \quad \forall r\in R \tag{13}$$

$$w^r_m \le B\sum_{(i,j)\in A} x^r_{ijm} \quad \forall r\in R, \forall m\in M^r \tag{14}$$

$$t^v_m \ge 0 \forall v\in V, \quad \forall m\in\{0\}\cup M^v \tag{15}$$

$$w^v_m \ge 0 \forall v\in V, \quad \forall m\in\{0\}\cup M^v \tag{16}$$

$$t^v_{m+1} = t^v_m + w^v_m + \sum_{(i,j)\in A} y^v_{ijm}\delta_{ij} \forall v\in V, \quad \forall m\in\{0\}\cup M^v\setminus\{\mathscr{M}_v\} \tag{17}$$

$$t^v_0 = 0 \quad \forall v\in V \tag{18}$$

$$t^v_{\mathscr{M}_v} \le T_{Max} \quad \forall v\in V \tag{19}$$

$$w^v_m \le B\sum_{(i,j)\in A} y^v_{ijm} \ \forall v\in V, \ \forall m\in M^r \tag{20}$$

$$d_r^+ \geq pd^r \quad \forall r \in R \tag{21}$$

$$d_r^+ \geq t_0^r + w_0^r \quad \forall r \in R \tag{22}$$

$$d_r^- \leq pd^r \quad \forall r \in R \tag{23}$$

$$d_r^- \leq t_0^r + w_0^r \quad \forall r \in R \tag{24}$$

$$c_r^+ \geq pa^r \quad \forall r \in R \tag{25}$$

$$c_r^+ \geq t_{\mathscr{M}_r}^r \quad \forall r \in R \tag{26}$$

$$c_r^- \leq pa^r \quad \forall r \in R \tag{27}$$

$$c_r^- \leq t_{\mathscr{M}_r}^r \quad \forall r \in R \tag{28}$$

$$\sum_{v\in V}\sum_{m_v\in M^v} p_{rm_rvm_v} \leq 1 \quad \forall r \in R, m_r \in M^r \tag{29}$$

$$\sum_{r\in R}\sum_{m_r\in M^r} p_{rm_rvm_v} q^r \leq q^v \quad \forall v \in V, m_v \in M^v \tag{30}$$

$$x_{ijm_r}^r \leq y_{ijm_v}^v + \left(1 - p_{rm_rvm_v}\right) \quad \forall v \in V, \forall r \in R, \forall (i,j) \in A, \forall m_r \in M^r, \forall m_v \in M^v \tag{31}$$

$$x_{ijm_r}^r \geq y_{ijm_v}^v - \left(1 - p_{rm_rvm_v}\right) \quad \forall v \in V, \forall r \in R, \forall (i,j) \in A, \forall m_r \in M^r, \forall m_v \in M^v \tag{32}$$

$$t_{m_r}^r \leq t_{m_v}^v + \left(1 - p_{rm_rvm_v}\right)B \quad \forall v \in V, \forall r \in R, \forall m_r \in M^r, \forall m_v \in M^v \tag{33}$$

$$t_{m_r}^r \geq t_{m_v}^v - \left(1 - p_{rm_rvm_v}\right)B \quad \forall v \in V, \forall r \in R, \forall m_r \in M^r, \forall m_v \in M^v \tag{34}$$

$$\sum_{(i,j)\in A} x_{ijm_r}^r \leq \sum_{v\in V}\sum_{m_v\in M^v} p_{rm_rvm_v} \quad \forall r \in R, \forall m_r \in M^r \tag{35}$$

$$Ba_{rv} \geq \sum_{m_r\in M^r}\sum_{m_v\in M^v} p_{rm_rvm_v} \quad \forall v \in V, \forall r \in R \tag{36}$$

$$\sum_{v\in V} a_{rv} - 1 \leq d^r \quad \forall r \in R \tag{37}$$

$$x_{ijm}^r \in \{0,1\} \quad \forall r \in R, \forall (i,j) \in A, \forall m \in M^r \tag{93}$$

$$y_{ijm}^v \in \{0,1\} \quad \forall v \in V, \forall (i,j) \in A, \forall m \in M^v \tag{94}$$

$$a_{rv} \in \{0,1\} \quad \forall r \in R, \forall v \in V \tag{95}$$

$$p_{rm_rvm_v} \in \{0,1\} \quad \forall r \in R, \forall m_r \in M^r, \forall v \in V, \forall m_v \in M^v \tag{96}$$

D.2 Continuous time model - Extension

$$\begin{aligned}\text{minimize } Z = {} & \alpha\sum_{v\in V}\sum_{(i,j)\in A}\sum_{m\in M^v} y_{ijm}^v l_{ij} + \beta\sum_{r\in R} q^r\left(t_{\mathscr{M}_r}^r - \sum_{m\in\{0\}\cup M^r} w_m^r - t_0^r\right) \\ & +\mu_1\sum_{r\in R}\left(pd^r - d_r^-\right) + \mu_2\sum_{r\in R}\left(d_r^+ - pd^r\right) + \mu_3\sum_{r\in R}\left(c_r^+ - pa^r\right) \\ & +\mu_4\sum_{r\in R}\left(pa^r - c_r^-\right) + \eta\sum_{r\in R}\sum_{v\in V} a_{rv} q^r + \gamma_1\sum_{r\in R}\sum_{m\in M^r} q^r w_m^r \\ & +\gamma_2\sum_{v\in V}\sum_{m\in\{0\}\cup M^v} w_m^v + E\sum_{r\in R} u^r q^r\end{aligned} \tag{1}$$

such that

$$\sum_{(i,j)\in A} x^r_{ijm} \leq 1 \quad \forall r \in R, \forall m \in M^r \tag{2}$$

$$\sum_{(o^r,j)\in A} x^r_{o^r j1} \geq \sum_{(i,j)\in A} x^r_{ij1} \quad \forall r \in R \tag{3}$$

$$\sum_{(i,j)\in A} x^r_{ijm} = \sum_{(j,k)\in A} x^r_{jk(m+1)} \quad \forall r \in R, \forall m \in M^r, j \neq d^r \tag{4}$$

$$u^r + \sum_{m\in M^r} \sum_{(i,d^r)\in A} x^r_{id^r m} = 1 \quad \forall r \in R \tag{5}$$

$$\sum_{(i,j)\in A} y^v_{ijm} \leq 1 \quad \forall v \in V, \forall m \in M^v \tag{6}$$

$$\sum_{(o^v,j)\in A} y^v_{o^v j1} \geq \sum_{(i,j)\in A} y^v_{ij1} \quad \forall v \in V \tag{7}$$

$$\sum_{(i,j)\in A} y^v_{ijm} \geq \sum_{(j,k)\in A} y^v_{jk(m+1)} \quad \forall v \in V, \forall m \in M^v \setminus \{\mathscr{M}_v\}, \forall j \in N \tag{8}$$

$$t^r_m \geq 0 \quad \forall r \in R, \forall m \in \{0\} \cup M^r \tag{9}$$

$$w^r_m \geq 0 \quad \forall r \in R, \forall m \in \{0\} \cup M^r \tag{10}$$

$$t^r_0 = e^r \quad \forall r \in R \tag{12}$$

$$t^r_{\mathscr{M}_r} \leq l^r \quad \forall r \in R \tag{13}$$

$$w^r_m \leq B \sum_{(i,j)\in A} x^r_{ijm} \quad \forall r \in R, \forall m \in M^r \tag{14}$$

$$t^v_m \geq 0 \quad \forall v \in V, \forall m \in \{0\} \cup M^v \tag{15}$$

$$w^v_m \geq 0 \quad \forall v \in V, \forall m \in \{0\} \cup M^v \tag{16}$$

$$t^v_0 = 0 \quad \forall v \in V \tag{18}$$

$$t^v_{\mathscr{M}_v} \leq T_{Max} \quad \forall v \in V \tag{19}$$

$$w^v_m \leq B \sum_{(i,j)\in A} y^v_{ijm} \quad \forall v \in V, \forall m \in M^r \tag{20}$$

$$d^+_r \geq pd^r \quad \forall r \in R \tag{21}$$

$$d^+_r \geq t^r_0 + w^r_0 \quad \forall r \in R \tag{22}$$

$$d^-_r \leq pd^r \quad \forall r \in R \tag{23}$$

$$d^-_r \leq t^r_0 + w^r_0 \quad \forall r \in R \tag{24}$$

$$c^+_r \geq pa^r \quad \forall r \in R \tag{25}$$

$$c^+_r \geq t^r_{\mathscr{M}_r} \quad \forall r \in R \tag{26}$$

$$c^-_r \leq pa^r \quad \forall r \in R \tag{27}$$

$$c^-_r \leq t^r_{\mathscr{M}_r} \quad \forall r \in R \tag{28}$$

$$\sum_{v \in V} \sum_{m_v \in M^v} p_{rm_r vm_v} \leq 1 \quad \forall r \in R, m_r \in M^r \tag{29}$$

$$\sum_{r \in R} \sum_{m_r \in M^r} p_{rm_r vm_v} q^r \leq q^v \quad \forall v \in V, m_v \in M^v \tag{30}$$

$$x^r_{ijm_r} \leq y^v_{ijm_v} + \left(1 - p_{rm_r vm_v}\right) \quad \forall v \in V, \forall r \in R, \forall (i,j) \in A, \forall m_r \in M^r, \forall m_v \in M^v \tag{31}$$

$$x^r_{ijm_r} \geq y^v_{ijm_v} - \left(1 - p_{rm_r vm_v}\right) \quad \forall v \in V, \forall r \in R, \forall (i,j) \in A, \forall m_r \in M^r, \forall m_v \in M^v \tag{32}$$

$$t^r_{m_r} \leq t^v_{m_v} + \left(1 - p_{rm_r vm_v}\right)B \quad \forall v \in V, \forall r \in R, \forall m_r \in M^r, \forall m_v \in M^v \tag{33}$$

$$t^r_{m_r} \geq t^v_{m_v} - \left(1 - p_{rm_r vm_v}\right)B \quad \forall v \in V, \forall r \in R, \forall m_r \in M^r, \forall m_v \in M^v \tag{34}$$

$$\sum_{(i,j) \in A} x^r_{ijm_r} \leq \sum_{v \in V} \sum_{m_v \in M^v} p_{rm_r vm_v} \quad \forall r \in R, \forall m_r \in M^r \tag{35}$$

$$Ba_{rv} \geq \sum_{m_r \in M^r} \sum_{m_v \in M^v} p_{rm_r vm_v} \quad \forall v \in V, \forall r \in R \tag{36}$$

$$\sum_{v \in V} a_{rv} - 1 \leq d^r \quad \forall r \in R \tag{37}$$

$$g^r_1 \geq \sum_{v \in V} \sum_{m_v \in M^v} p_{r1vm_v}, \quad \forall r \in R \tag{38}$$

$$g^r_{m_r} \geq \sum_{v \in V} p_{rm_r v1}, \forall r \in R, \quad \forall m_r \in M^r \tag{39}$$

$$g^r_{m_r} \geq p_{rm_r vm_v} - p_{rm_r-1vm_v-1} \quad \forall v \in V, \forall m_v \in M^v \setminus \{1\}, \forall r \in R, \forall m_r \in M^r \setminus \{1\} \tag{40}$$

$$\sum_{r \in R} h_{vmr} \leq 1 \quad \forall v \in V, \forall m \in M^v \tag{42}$$

$$\sum_{r \in R} h_{v1r} s_r \geq p_{rm_r v1} s_r \quad \forall v \in V, \forall r \in R, \forall m_r \in M^r \tag{43}$$

$$\sum_{r \in R} h_{vm_v r} s_r \geq p_{r1vm_v} s_r \quad \forall v \in V, \forall m_v \in M^v, \forall r \in R \tag{44}$$

$$\sum_{r \in R} h_{vm_v r} s_r \geq \left(p_{rm_r vm_v} - p_{rm_r-1vm_v-1}\right) s_r \quad \forall v \in V, \forall m_v \in M^v \setminus \{1\}, \forall r \in R, \forall m_r \in M^r \setminus \{1\} \tag{45}$$

$$h_{vm_v r} \leq 1 - \sum_{m_r \in M^r} p_{rm_r vm_v-1} \quad \forall v \in V, \forall m_v \in M^v \setminus \{1\}, \forall r \in R \tag{46}$$

$$h_{vm_v r} \leq \sum_{m_r \in M^r} p_{rm_r vm_v} \quad \forall v \in V, \forall m_v \in M^v, \forall r \in R \tag{47}$$

$$t^r_{m_r-1} + w^r_{m_r-1} \leq t^v_{m_v-1} + w^v_{m_v-1} + \left(1 - p_{rm_r vm_v}\right)B \quad \forall v \in V, \forall r \in R, \forall m_r \in M^r, \forall m_v \in M^v \tag{49}$$

$$t^r_{m_r-1} + w^r_{m_r-1} \geq t^v_{m_v-1} + w^v_{m_v-1} - \left(1 - p_{rm_r vm_v}\right)B \quad \forall v \in V, \forall r \in R, \forall m_r \in M^r, \forall m_v \in M^v \tag{50}$$

$$\sum_{k \in TS} TT^r_{ijmk} = x^r_{ijm} \quad \forall r \in R, \forall m \in M^r, \forall (i,j) \in A \tag{51}$$

$$\sum_{(i,j) \in A} TT^r_{ijmk} lb_{ijk} \leq t^r_m + w^r_m + g^r_m s^r \quad \forall r \in R, \forall m \in M^r, \forall k \in TS \tag{52}$$

$$t^r_m + w^r_m + g^r_m s^r \leq \sum_{(i,j) \in A} TT^r_{ijmk} ub_{ijk} + B\left(1 - \sum_{(i,j) \in A} TT^r_{ijmk}\right) \quad \forall r \in R, \forall m \in M^r, \forall k \in TS \tag{53}$$

$$\sum_{k\in TS} TT^v_{ijmk} = y^v_{ijm} \quad \forall v \in V, \forall m \in M^v, \forall (i,j) \in A \tag{54}$$

$$\sum_{(i,j)\in A} TT^v_{ijmk} lb_{ijk} \le t^v_m + w^v_m + \sum_{r\in R} s^r h_{vmr} \quad \forall v \in V, \forall m \in M^v, \forall k \in TS \tag{55}$$

$$t^v_m + w^v_m + \sum_{r\in R} s^r h_{vmr} \le \sum_{(i,j)\in A} TT^v_{ijmk} ub_{ijk} + B\left(1 - \sum_{(i,j)\in A} TT^v_{ijmk}\right) \quad \forall v \in V, \forall m \in M^v, \forall k \in TS \tag{56}$$

$$t^r_{m+1} = t^r_m + w^r_m + g^r_m s^r + \sum_{k\in TS}\sum_{(i,j)\in A} TT^r_{ijmk}\delta_{ijk} \quad \forall r \in R, \forall m \in M^r \setminus \{\mathscr{M}^r\} \tag{57}$$

$$t^v_{m+1} = t^v_m + w^v_m + \sum_{r\in R} s^r h_{vmr} + \sum_{k\in TS}\sum_{(i,j)\in A} TT^v_{ijmk}\delta_{ijk} \quad \forall v \in V, \forall m \in M^v \setminus \{\mathscr{M}^v\} \tag{58}$$

$$x^r_{ijm} \in \{0,1\} \;\; \forall r \in R, \;\; \forall (i,j) \in A, \forall m \in M^r \tag{93}$$

$$y^v_{ijm} \in \{0,1\} \quad \forall v \in V, \forall (i,j) \in A, \forall m \in M^v \tag{94}$$

$$a_{rv} \in \{0,1\} \quad \forall r \in R, \forall v \in V \tag{95}$$

$$p_{rm_r vm_v} \in \{0,1\} \quad \forall r \in R, \forall m_r \in M^r, \forall v \in V, \forall m_v \in M^v \tag{96}$$

$$g^r_m \in \{0,1\} \quad \forall r \in R, \forall m \in M^r \tag{97}$$

$$h_{vmr} \in \{0,1\} \quad \forall r \in R, \forall v \in V, \forall m \in M^v \tag{98}$$

D.3 Discrete time model - Core

$$\begin{aligned}\text{minimize } Z = {} & \alpha\sum_{v\in V}\sum_{(i,j,t)\in A^*} y^v_{ijt} l_{ij} + \beta\sum_{r\in R}\sum_{(i,j,t)\in A^*, i\ne j} x^r_{ijt}\delta_{ijt} q^r \\ & +\mu_1\sum_{r\in R}\sum_{t<pd^r\in T^r}\left(1-x^r_{o^r o^r t}\right)q^r + \mu_2\sum_{r\in R}\sum_{t\ge pd^r\in T^r} x^r_{o^r o^r t} q^r \\ & +\mu_3\sum_{r\in R}\sum_{t<pa^r\in T^r} x^r_{d^r d^r t} q^r + \mu_4\sum_{r\in R}\sum_{t\ge pa^r\in T^r}\left(1-x^r_{d^r d^r t}\right)q^r \\ & +\eta\sum_{r\in R}\sum_{v\in V} a_{rv} q^r + \gamma_1\sum_{r\in R}\sum_{(i,i,t)\in A^*} x^r_{iit} q^r + \gamma_2\sum_{v\in V}\sum_{(i,i,t)\in A^*} y^v_{iit} + E\sum_{r\in R} u^r q^r\end{aligned} \tag{59}$$

such that:

$$\sum_{i,j\in N, i\ne o^r} x^r_{ije^r} = 0 \quad \forall r \in R \tag{60}$$

$$\sum_{j\in N} x^r_{o^r j e^r} = 1 \quad \forall r \in R \tag{61}$$

$$\sum_{i\in N} x^r_{id^r t_2} + u^r = 1 \quad \forall r \in R, t_2 \in T^r | t_2 + \delta_{id^r t_2} = l^r \tag{62}$$

$$\sum_{i,j\in N, i\ne o^v} y^v_{ij0} = 0 \quad \forall v \in V \tag{63}$$

$$\sum_{j\in N} y^v_{o^v j0} = 1 \quad \forall v \in V \tag{64}$$

$$\sum_{i\in N}\sum_{j\in N} y^v_{ijt} = 1 \quad \forall v \in V, \forall t \in T \middle| t + \delta_{ijt} = T_{Max} \tag{65}$$

$$\sum_{i\in N} x^r_{ijt_2} = \sum_{i\in N} x^r_{jit} \quad \forall r \in R, \forall t \in T^r, t_2 \in T^r \middle| t_2 + \delta_{ijt_2} = t, \forall j \in N \tag{66}$$

$$\sum_{i\in N} y^v_{ijt_2} = \sum_{i\in N} y^v_{jit} \quad \forall v\in V, \forall j\in N, \forall t\in T/\{T^{max}\}, t_2\in T \big| t_2+\delta_{ijt_2} = t \tag{67}$$

$$\sum_{v\in V} a_{rvt} \le 1 \quad \forall r\in R, \forall t\in T^r \tag{68}$$

$$\sum_{r\in R} a_{rvt} q^r \le q^v \quad \forall v\in V, \forall t\in T \tag{69}$$

$$\sum_{j\in N} x^r_{ijt} \le \sum_{j\in N} y^v_{ijt} + (1-a_{rvt}) \quad \forall r\in R, \forall v\in V, \forall i\in N, \forall t\in T^r \tag{72}$$

$$\sum_{j\in N} x^r_{ijt} \ge \sum_{j\in N} y^v_{ijt} - (1-a_{rvt}) \quad \forall r\in R, \forall v\in V, \forall i\in N, \forall t\in T^r \tag{73}$$

$$\sum_{i\in N} x^r_{ijt} \le \sum_{i\in N} y^v_{ijt} + (1-a_{rvt}) \quad \forall r\in R, \forall v\in V, \forall j\in N, \forall t\in T^r \tag{74}$$

$$\sum_{i\in N} x^r_{ijt} \ge \sum_{i\in N} y^v_{ijt} - (1-a_{rvt}) \quad \forall r\in R, \forall v\in V, \forall j\in N, \forall t\in T^r \tag{75}$$

$$\sum_{i\in N} x^r_{iit} \ge 1 - \sum_{v\in V} a_{rvt} \quad \forall r\in R, \forall t\in T^r \tag{76}$$

$$Ba_{rv} \ge \sum_{t\in T} a_{rvt} \quad \forall v\in V, \forall r\in R \tag{77}$$

$$\sum_{v\in V} a_{rv} - 1 \le d^r \quad \forall r\in R \tag{78}$$

$$x^r_{ijt} \in \{0,1\} \quad \forall r\in R, \forall (i,j,t)\in A^* \tag{99}$$

$$y^v_{ijt} \in \{0,1\} \quad \forall v\in V, \forall (i,j,t)\in A^* \tag{100}$$

$$a_{rv} \in \{0,1\} \quad \forall r\in R, \forall v\in V \tag{101}$$

$$a_{rvt} \in \{0,1\} \quad \forall r\in R, \forall v\in V, \forall t\in T^r \tag{102}$$

D.4 Discrete time model - Extension

$$\begin{aligned}\text{minimize } Z = &\alpha\sum_{v\in V}\sum_{(i,j,t)\in A^*} y^v_{ijt} l_{ij} + \beta\sum_{r\in R}\sum_{(i,j,t)\in A^*, i\neq j} x^r_{ijt}\delta_{ijt} q^r \\ &+\mu_1\sum_{r\in R}\sum_{t<(pd^r+s^r)\in T^r}\left(1-x^r_{o^r o^r t}\right)q^r + \mu_2\sum_{r\in R}\sum_{t\ge(pd^r+s^r)\in T^r} x^r_{o^r o^r t} q^r \\ &+\mu_3\sum_{r\in R}\sum_{t<pa^r\in T^r} x^r_{d^r d^r t} q^r + \mu_4\sum_{r\in R}\sum_{t\ge pa^r\in T^r}\left(1-x^r_{d^r d^r t}\right)q^r \\ &+\eta\sum_{r\in R}\sum_{v\in V} a_{rv} q^r + \gamma_1\sum_{r\in R}\sum_{(i,i,t)\in A^*} x^r_{iit} q^r + \gamma_2\sum_{v\in V}\sum_{(i,i,t)\in A^*} y^v_{iit} + E\sum_{r\in R} u^r q^r\end{aligned} \tag{59}$$

such that:

$$\sum_{i,j\in N, i\neq o^r} x^r_{ije^r} = 0 \quad \forall r\in R \tag{60}$$

$$\sum_{j\in N} x^r_{o^r j e^r} = 1 \quad \forall r\in R \tag{61}$$

$$\sum_{i\in N} x^r_{id^r t_2} + u^r = 1 \quad \forall r\in R, t_2\in T^r | t_2 + \delta_{id^r t_2} = l^r \tag{62}$$

$$\sum_{i,j \in N, i \neq o^v} y^v_{ij0} = 0 \quad \forall v \in V \tag{63}$$

$$\sum_{j \in N} y^v_{o^v j0} = 1 \quad \forall v \in V \tag{64}$$

$$\sum_{i \in N} \sum_{j \in N} y^v_{ijt} = 1 \quad \forall v \in V, \forall t \in T \middle| t + \delta_{ijt} = T_{Max} \tag{65}$$

$$\sum_{i \in N} x^r_{ijt_2} = \sum_{i \in N} x^r_{jit} \quad \forall r \in R, \forall t \in T^r, t_2 \in T^r \middle| t_2 + \delta_{ijt_2} = t, \forall j \in N \tag{66}$$

$$\sum_{i \in N} y^v_{ijt_2} = \sum_{i \in N} y^v_{jit} \quad \forall v \in V, \forall j \in N, \forall t \in T / \{T^{max}\}, t_2 \in T \middle| t_2 + \delta_{ijt_2} = t \tag{67}$$

$$\sum_{v \in V} a_{rvt} \leq 1 \quad \forall r \in R, \forall t \in T^r \tag{68}$$

$$\sum_{r \in R} a_{rvt} q^r \leq q^v \quad \forall v \in V, \forall t \in T \tag{69}$$

$$\sum_{j \in N} x^r_{ijt} \leq \sum_{j \in N} y^v_{ijt} + (1 - a_{rvt}) \quad \forall r \in R, \forall v \in V, \forall i \in N, \forall t \in T^r \tag{72}$$

$$\sum_{j \in N} x^r_{ijt} \geq \sum_{j \in N} y^v_{ijt} - (1 - a_{rvt}) \quad \forall r \in R, \forall v \in V, \forall i \in N, \forall t \in T^r \tag{73}$$

$$\sum_{i \in N} x^r_{ijt} \leq \sum_{i \in N} y^v_{ijt} + (1 - a_{rvt}) \quad \forall r \in R, \forall v \in V, \forall j \in N, \forall t \in T^r \tag{74}$$

$$\sum_{i \in N} x^r_{ijt} \geq \sum_{i \in N} y^v_{ijt} - (1 - a_{rvt}) \quad \forall r \in R, \forall v \in V, \forall j \in N, \forall t \in T^r \tag{75}$$

$$\sum_{i \in N} x^r_{iit} \geq 1 - \sum_{v \in V} a_{rvt} \quad \forall r \in R, \forall t \in T^r \tag{76}$$

$$B a_{rv} \geq \sum_{t \in T} a_{rvt} \quad \forall v \in V, \forall r \in R \tag{77}$$

$$\sum_{v \in V} a_{rv} - 1 \leq d^r \quad \forall r \in R \tag{78}$$

$$\sum_{i \in N} x^r_{iit} \geq a_{rvt} - a_{rvt-s^r} \quad \forall t \in T^r | t - s^r \in T^r, \forall r \in R, \forall v \in V \tag{79}$$

$$x^r_{ijt} \in \{0,1\} \quad \forall r \in R, \forall (i,j,t) \in A^* \tag{99}$$

$$y^v_{ijt} \in \{0,1\} \quad \forall v \in V, \forall (i,j,t) \in A^* \tag{100}$$

$$a_{rv} \in \{0,1\} \quad \forall r \in R, \forall v \in V \tag{101}$$

$$a_{rvt} \in \{0,1\} \quad \forall r \in R, \forall v \in V, \forall t \in T^r \tag{102}$$

References

Agatz, N., Erera, A., Savelsbergh, M., Wang, X., 2012. Optimization for dynamic ride-sharing: a review. Eur. J. Oper. Res. 223 (2), 295–303. http://dx.doi.org/10.1016/j.ejor.2012.05.028. URL.

Conceição, L., Correia, G.H., Tavares, J.P., 2017. The deployment of automated vehicles in urban transport systems: a methodology to design dedicated zones. Transport. Resear. Procedia 27, 230–237. http://dx.doi.org/10.1016/j.trpro.2017.12.025, 20th EURO Working Group on Transportation Meeting, EWGT 2017, 4-6 September 2017, Budapest, Hungary. URL.

Cordeau, J.-F., Laporte, G., 2003. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. Transp. Res. Part B Methodol. 37 (6), 579–594. http://dx.doi.org/10.1016/S0191-2615(02)00045-0. URL.

Correia, G.H., van Arem, B., 2016. 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. http://dx.doi.org/10.1016/j.trb.2016.03.002. URL.

Cortés, C.E., Matamala, M., Contardo, C., 2010. The pickup and delivery problem with transfers: formulation and a branch-and-cut solution method. Eur. J. Oper. Res. 200 (3), 711–724. http://dx.doi.org/10.1016/j.ejor.2009.01.022. URL.

Danloup, N., Allaoui, H., Goncalves, G., 2018. A comparison of two meta-heuristics for the pickup and delivery problem with transshipment. Comput. Oper. Res. 100, 155–171. http://dx.doi.org/10.1016/j.cor.2018.07.013. URL.

Deleplanque, S., Quilliot, A., Aug. 2013. Transfers in the on-demand transportation: the DARPT Dial-a-Ride Problem with transfers allowed. In: Multidisciplinary International Scheduling Conference: Theory and Applications (MISTA). Vol. ISSN 2305-249X of the MISTA Conference Series. Ghent, Belgium, pp. 185–205. URL. https://hal.archives-ouvertes.fr/hal-00917197/file/MISTRA.DARPT.Final.pdf.

Donati, A.V., Montemanni, R., Casagrande, N., Rizzoli, A.E., Gambardella, L.M., 2008. Time dependent vehicle routing problem with a multi ant colony system. Eur. J. Oper. Res. 185 (3), 1174–1191. http://dx.doi.org/10.1016/j.ejor.2006.06.047. URL.

Fagnant, D.J., Kockelman, K., 2015. Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations. Transport. Res. Pol. Pract. 77, 167–181. http://dx.doi.org/10.1016/j.tra.2015.04.003. URL.

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

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

Hou, Y., Zhong, W., Su, L., Hulme, K., Sadek, A.W., Qiao, C., Dec 2016. Taset: improving the efficiency of electric taxis with transfer-allowed rideshare. IEEE Trans. Veh. Technol. 65 (12), 9518–9528. http://dx.doi.org/10.1109/TVT.2016.2592983. URL.

Ichoua, S., Gendreau, M., Potvin, J.-Y., 2003. Vehicle dispatching with time-dependent travel times. Eur. J. Oper. Res. 144 (2), 379–396. http://dx.doi.org/10.1016/S0377-2217(02)00147-9. URL.

International Transport Forum, t., 2015. Urban mobility system upgrade (6). URL. https://doi.org/10.1787/5jlwvzdk29g5-en.

Liang, X., Correia, G.H., van Arem, B., 2017. An optimization model for vehicle routing of automated taxi trips with dynamic travel times. In: 20th EURO Working Group on Transportation Meeting, EWGT 2017, 4-6 September 2017, Budapest, Hungary, vol. 27, pp. 736–743. http://dx.doi.org/10.1016/j.trpro.2017.12.045. URL.

Liu, J., Kockelman, K.M., Boesch, P.M., Ciari, F., Nov, 2017. Tracking a system of shared autonomous vehicles across the Austin, Texas network using agent-based simulation. Transportation 44 (6), 1261–1278. http://dx.doi.org/10.1007/s11116-017-9811-1. URL.

Malandraki, C., Daskin, M.S., 1992. Time dependent vehicle routing problems: formulations, properties and heuristic algorithms. Transport. Sci. 26 (3), 185–200. http://dx.doi.org/10.1287/trsc.26.3.185. URL.

Martinez, L.M., Correia, G.H., Viegas, J.M., 2015. An agent-based simulation model to assess the impacts of introducing a shared-taxi system: an application to Lisbon (Portugal). J. Adv. Transport. 49 (3), 475–495. http://dx.doi.org/10.1002/atr.1283. URL.

Masson, R., Lehuédé, F., Péton, O., 2013. An adaptive large neighborhood search for the pickup and delivery problem with transfers. Transport. Sci. 47 (3), 344–355. http://dx.doi.org/10.1287/trsc.1120.0432. URL.

Masson, R., Lehuédé, F., Péton, O., 2014. The dial-a-ride problem with transfers. Comput. Oper. Res. 41, 12–23. http://dx.doi.org/10.1016/j.cor.2013.07.020. URL.

Molenbruch, Y., Braekers, K., Caris, A., Dec 2017. Typology and literature review for dial-a-ride problems. Ann. Oper. Res. 259 (1), 295–325. URL 10.1007/s10479-017-2525-0.

Peng, Z., Al Chami, Z., Manier, H., Manier, M.A., 2019. A hybrid particle swarm optimization for the selective pickup and delivery problem with transfers. Eng. Appl. Artif. Intell. 85, 99–111. http://dx.doi.org/10.1016/j.engappai.2019.06.006. URL.

Petersen, H.L., Ropke, S., 2011. The pickup and delivery problem with cross-docking opportunity. In: Böse, J.W., Hu, H., Jahn, C., Shi, X., Stahlbock, R., Voß, S. (Eds.), Computational Logistics. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 101–113. http://dx.doi.org/10.1007/978-3-642-24264-9_8. URL.

Posada, M., Andersson, H., Häll, C.H., Jul, 2017. The integrated dial-a-ride problem with timetabled fixed route service. Publ. Transport 9 (1), 217–241. http://dx.doi.org/10.1007/s12469-016-0128-9. URL.

Rais, A., Alvelos, F., Carvalho, M., 2014. New mixed integer-programming model for the pickup-and-delivery problem with transshipment. Eur. J. Oper. Res. 235 (3), 530–539. http://dx.doi.org/10.1016/j.ejor.2013.10.038. URL.

Reinhardt, L.B., Clausen, T., Pisinger, D., 2013. Synchronized dial-a-ride transportation of disabled passengers at airports. Eur. J. Oper. Res. 225 (1), 106–117. http://dx.doi.org/10.1016/j.ejor.2012.09.008. URL.

Schmid, V., Doerner, K.F., 2010. Ambulance location and relocation problems with time-dependent travel times. Eur. J. Oper. Res. 207 (3), 1293–1303. http://dx.doi.org/10.1016/j.ejor.2010.06.033. URL.

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. In: Road Vehicle Automation. Springer, pp. 229–245. http://dx.doi.org/10.1007/978-3-319-05990-7_20. URL.

Thangiah, S.R., Fergany, A., Awan, S., Nov, 2007. Real-time split-delivery pickup and delivery time window problems with transfers. Cent. Eur. J. Oper. Res. 15 (4), 329–349. http://dx.doi.org/10.1007/s10100-007-0035-x. URL.

van den Berg, P.L., van Essen, J.T., 2019. Scheduling non-urgent patient transportation while maximizing emergency coverage. Transport. Sci. 53 (2), 492–509. http://dx.doi.org/10.1287/trsc.2018.0823. URL.

van den Bergh, J., Quttineh, N.-H., Larsson, T., Beliën, J., 2016. A time-indexed generalized vehicle routing model for military aircraft mission planning. In: Lübbecke, M., Koster, A., Letmathe, P., Madlener, R., Peis, B., Walther, G. (Eds.), Operations Research Proceedings 2014. Springer International Publishing, Cham, pp. 605–611. http://dx.doi.org/10.1007/978-3-319-28697-6_84. URL.