Open-Source Public Transportation Mobility Simulation Engine DTALite-S: A Discretized Space–Time Network-Based Modeling Framework for Bridging Multi-agent Simulation and Optimization

Lu Tong^{1,2} · Yuyan Pan³ · Pan Shang⁴ · Jifu Guo⁵ · Kai Xian⁵ · Xuesong Zhou^{5,6}

Received: 24 November 2018 / Revised: 8 December 2018 / Accepted: 17 December 2018 / Published online: 21 January 2019
© The Author(s) 2019

Abstract Recently, an open-source light-weight dynamic traffic assignment (DTA) package, namely DTALite, has been developed to allow a rapid utilization of advanced dynamic traffic analysis capabilities. Aiming to bridge the modeling gaps between multi-agent simulation and optimization in a multimodal environment, we further design and develop DTALite-S to simplify the traffic flow

dynamic representation details in DTALite for future extensions. We hope to offer a unified modeling framework with inherently consistent space–time network representations for both optimization formulation and simulation process. This paper includes three major modeling components: (1) mathematic formulations to describe traffic and public transportation simulation problem on a space–time network; (2) transportation transition dynamics involving multiple agents in the optimization process; (3) an alternating direction method of multipliers (ADMM)-based modeling structure to link different features between multi-agent simulation and optimization used in transportation. This unified framework can be embedded in a Lagrangian relaxation method and a time-oriented sequential simulation procedure to handle many general applications. We carried out a case study by using this unified framework to simulate the passenger traveling process in Beijing subway network which contains 18 urban rail transit lines, 343 stations, and 52 transfer stations. Via the ADMM-based solution approach, queue lengths at platforms, in-vehicle congestion levels and absolute deviation of travel times are obtained within 1560 seconds. The case study indicates that the open-source DTALite-S integrates simulation and optimization procedure for complex dynamic transportation systems and can efficiently generate comprehensive space-time traveling status.

Keywords Space–time network · Dynamic traffic assignment · Multi-agent simulation · Lagrangian relaxation · Alternating direction method of multipliers

✉ Yuyan Pan
panyuyan8879@sina.com

Lu Tong
ltong@buaa.edu.cn

Pan Shang
shangp16@mails.tsinghua.edu.cn

Jifu Guo
guojf@bjtrc.org.cn

Kai Xian
xiank@bjtrc.org.cn

Xuesong Zhou
xzhou74@asu.edu

- ¹ School of Electronic and Information Engineering, Beihang University, Beijing, China
- ² National Engineering Laboratory for Comprehensive Transportation Big Data Application Technology, Beijing, China
- ³ Beijing University of Technology, Beijing, China
- ⁴ Department of Civil Engineering, Tsinghua University, Beijing, China
- ⁵ Beijing Transport Institute, No. 9 LiuLiQiao South Road, Fengtai District, Beijing, China
- ⁶ School of Sustainable Engineering and the Built Environment, Arizona State University, Tempe, USA

Communicated by Lixing Yang.

1 Introduction

To understand and analyze future emerging mobility scenarios, planners and engineers need to utilize many different simulation tools to generate corresponding modeling results. The main purpose of transportation simulation is to shed more light on the underlying mechanisms or potential problems that control the behavior of a complex transportation system.

Typically, simulating a system involves a probabilistic input model, a set of dynamic equations or constraints between the inputs and outputs, and then finally produces a set of outputs under different input instances. Optimization, on the other hand, needs to search a solution in the dynamic (possibly complex) system subject to a number of constraints. There are a wide range of studies focusing on simulation-based optimization, to name a few, a leading study by Osorio et al. [1] involving stochastic urban traffic simulators, and another study by Xiong et al. [2] using the DTALite simulator. Generally, transportation planners and engineers utilize simulation tools to evaluate and further optimize a subset of system's parameters, but there is a critical modeling gap between simulation and optimization for complex dynamic transportation systems. To bridge such a gap in a multimodal environment, this research focuses on how to offer a theoretically sound and practically useful modeling framework with a simplified traffic flow dynamic.

1.1 Literature Review

Scheduling vehicles on congested transportation networks needs to consider both traffic flows with vehicle-to-road assignment and vehicle routing problem (VRP) with passenger-to-vehicle matching. There are numbers of studies about agent-based traffic assignment and traffic simulation. Mahmassani et al. [3] used flow-density relationships to predict time-dependent traffic flows in the Dynamic Network Assignment-Simulation Model for Advanced Roadway Telematics (DYNASMART). From a broader multi-agent optimization perspective, in the study by Nedic et al. [4], a distributed computation model is built for optimizing a sum of convex objective functions for all types of agents. For shared autonomous vehicle (SAV) operating, Fagnant et al. [5] proposed an agent-based shared autonomous vehicle relocation model in order to reduce potential users' wait times. Following Newell's kinematic approach [6], Zhou and Taylor [7] designed a mesoscopic traffic simulation approach and developed a time-driven open-source traffic assignment package DTALite to simulate large-scale networks with millions of vehicles. Based on the multi-source data generated from transportation network companies, Spieser et al. [8] provide an on-demand transportation

service model and estimate the benefits of sharing vehicles. Focusing on modeling the microscopic behavior in virtual reality systems, Yu et al. [9] provided a hierarchical modular modeling and distributed simulation methodology. A concise overview of simulation-based transportation analysis approaches is offered by Bierlaire [10].

Transportation researchers have devoted significant attentions to both traffic and public transportation simulation models. Recently, Bradley et al. [11] conducted possible autonomous vehicle (AV) operating scenarios in a road network system, and further modeled the metro transit station as a finite capacity queuing system through a discrete-event simulation (DES) approach, which was also adopted in the study by Afaq et al. [12]. Liang et al. [13] provided a mathematical model to consider the door-to-door intermodal travel trips and found that the vehicle fleet size directly influences the performance of the taxi system. Mahmassani [14] integrated varying behavioral mechanisms for different classes of vehicles into a microsimulation framework through a series of experiments under varying market penetration rates of AVs and/or connected vehicles. Qu et al. [15] presented a computationally efficient parallel-computing framework for real-life traffic simulation for metropolitan areas. To meet simulation accuracy requirements, Martinez et al. [16] proposed an agent-based model to simulate individual daily mobility while traffic assignment conditions are updated every 5 min. Golubev et al. [17] presented an agent-based traffic modeling framework allowing users to set a specific model for each supported class. Sun et al. [18] presented an agent-based simulation for urban rail transit systems. Based on kinematic wave model, Wen et al. [19] implemented a shared autonomous mobility-on-demand (AMoD) modeling platform for simulating individual travelers and vehicles with demand–supply interaction and analyzing the system performance through various metrics of indicators.

Recently, there are many papers focusing on vehicle routing optimization models and algorithms used in large-scale optimization. Boyd et al. [20] discussed general distributed optimization and provided efficient implementation under the non-convex setting. Mahmoudi and Zhou [21] built the state-space–time network to model the vehicle routing problem with pickup and drop-off and with time windows (VRPPDTW). Based on the Lagrangian decomposition and space–time windows, Tong et al. [22] developed a joint optimization approach for customized bus services. Wei et al. [23] developed a set of integer programming and dynamic programming models to optimize simplified multi-vehicle trajectories. Zhou et al. [24] introduced a vehicle routing optimization engine VRPLite on the basis of a hyper space–time–state network representation with an embedded column generation and Lagrangian relaxation framework. Zhao et al. [25]

considered an optimization framework for electric vehicles in the one-way carsharing system, and they proposed a Lagrangian relaxation-based solution approach to decompose the primal problem.

1.2 Paper Structure

The remainder of this paper is organized as follows. The next section presents a modeling framework within a space–time transportation network representation. In Sect. 3, we formulate a set of space–time network-based formulations to describe the public transportation optimization problem. Section 4 provides a simulation process of vehicular loading with pickup/drop-off services based on Lagrangian relaxation. In Sect. 5, we use ADMM algorithm to expound the linkage between multi-agent-based optimization and simulation. Numerical experiments based on Beijing subway network are conducted in Sect. 6. In conclusion, Sect. 7 provides concluding remarks and future research work.

2 Problem Statement and Modeling Framework

2.1 Motivation

Many disruptive transportation technologies such as autonomous vehicles (AVs) and mobility as a service (MaaS) create new opportunities for public transit to provide high-level services. Along this line, transit operating researchers need to address some theoretical questions, including (a) how to consider individual system dynamics

in the context of public transportation network optimization, (b) how to use agent-based simulation tools to evaluate travel schedules with limited road capacity, and (c) how to improve system flexibility and accessibility with a consolidation of modes. To meet these challenges, this paper considers the vehicle route scheduling optimization in the broader framework of public transportation and develops a unified space–time framework to improve public transportation mobility. Table 1 gives an overarching list for different stakeholders and their measures to be examined in our analytical models.

Among widely used traffic simulation tools, DTALite is a queue-based mesoscopic traffic simulator, documented in the paper by Zhou and Taylor [7]. It is an open-source mesoscopic DTA (dynamic traffic assignment) simulation package designed to provide transportation planners, engineers, and researchers with a theoretically rigorous and computationally efficient traffic network modeling tool.

The transportation mobility simulation engine DTALite-S to be proposed in this paper is an important extension based on DTALite, with much more consistent representations for optimization in a space–time network. This fully functional, open-source DTALite-S can be downloaded from https://github.com/xzhou99/DTALite-S/tree/master/Version2_PickupDropoffSimulation/DTALite-S. In general, the software suite of DTALite-S aims to:

1. Provide an open-source simulation package that enables transportation researchers and students to understand the complex space–time network modeling process.

Table 1 Interested stakeholders and different measures that can be optimized and simulated

Stakeholders	Stakeholder interests	System-wide metrics	Individual user-oriented performance
(I) (Non-MaaS) users (passenger/drivers)	<ul style="list-style-type: none"> • Travel driving cost • Level of service 	<ul style="list-style-type: none"> • Total driving time • Congestion 	<ul style="list-style-type: none"> • Waiting time at bottlenecks • Mean travel time • Travel time reliability
(II) MaaS passengers	<ul style="list-style-type: none"> • MaaS service cost • Accessibility 	<ul style="list-style-type: none"> • Total end-to-end service time • MaaS service availability 	<ul style="list-style-type: none"> • Waiting time for MaaS vehicles • Service time window and fare • Detour factor for shared ride • Punctuality
(III) MaaS operator	<ul style="list-style-type: none"> • Operating and transportation costs • Revenue and profit from fares • Fleet structure and size of fleet 	<ul style="list-style-type: none"> • Revenue • Ridership • Punctuality • Total travel time 	<ul style="list-style-type: none"> • Vehicle traveling distance and travel time • Distance-based average passenger load • Cost per passenger served
(IV) Public transportation planning and management agencies	<ul style="list-style-type: none"> • System efficiency • Public equity • Sustainability 	<ul style="list-style-type: none"> • Space–time time accessibility • Motorized traffic • Non-motorized trips 	<ul style="list-style-type: none"> • Vehicle traveling distance • MaaS Service rate • Market share of green and shared transportation mode

2. Offer a unified framework with pickup and drop-off events that cover different traveling activities from driving-only to multiple public transportation modes, across the emerging transportation mobility spectrum, e.g., urban rail transit, synchronized bus, ride-sharing applications, as well as freight transport.
3. Agent-based dynamic traffic assignment and traffic simulation are integrated and extended to tackle practical applications of vehicle routing problem (VRP), or its variants, e.g., vehicle routing problem with pickup and delivery (VRPPD), vehicle routing problem with pickup and delivery with time windows (VRPPDTW).

2.2 Overall Modeling Framework

As shown in Fig. 1, there are five different elements in the overall modeling framework. The first category is **(1) input data**, covering a transportation network, passenger information with origin, destination, preferred departure/arrival time window and desired travel path; vehicle information with depot, time window, and capacity constraint. We consider two different types of vehicles, **(I)** the background vehicles, driven by their own drivers as single-occupancy vehicle (SOV) or high-occupancy vehicle (HOV), move along the given physical paths from the traffic assignment program, and **(III)** the MaaS vehicles operated by MaaS companies with optimized routes to meet the demand of MaaS passengers as stakeholder **(II)**. For the benefits of stakeholder **(IV)**, we have two coupled optimization modules, namely **(2) space-time network-based traffic assignment for background vehicles** to find the shortest path for each agent, **(3) transportation mobility optimization for passenger and vehicle in MaaS**, which assigns the tasks of serving passengers to specific vehicles

in a cost-optimal way, and finally, **(4) traffic simulation** for calculating traffic states in the transportation network. The system output **(5)** includes **time-dependent travel times**, and vehicle and passenger **space-time trajectories**. The traffic congestion factors, if need to reach the steady states or fixed point equilibrium, should be returned back to part of **(1) input data** for further consistent optimization and simulation. Please note that Roman numerals (I–IV) correspond to the stakeholders' indices in Table 1.

3 Mathematical Programming Model in a Disseized Network

3.1 Space-Time Network Construction for a Point Queue Model with Constant Capacity

Consider a physical transportation network $M = (N, L)$ with a finite set of nodes N and a finite set of links L , where nodes $i, j \in N$ and directed link $(i, j) \in L$. Set L is further divided into two subsets, i.e., passenger link set L^p and vehicle link set L^v . In this research, the physical transportation network is expanded into two coupled high-dimensional space-time networks, for passengers and vehicles. Besides typical space-time traveling/waiting arcs, additional space-time arcs, such as pickup and drop-off space-time arc, are also constructed for modeling the vehicle-passenger service process.

Let $G = (E, A)$ be a space-time network, where E is the set of space-time vertexes and $A = A^p \cup A^v \cup A^{PK}$ is the set of space-time links/arcs under the planning time horizon T . Each arc $(i, j, t, t') \in A$ indicates a directed space-time path from node i departing at time t to node j arriving at time t' . Thanks to the discretized space-time network structure, it is easy to incorporate passengers' travel request

Fig. 1 Flowchart with major variables in DTALite-S

Fig. 2 Vehicle trajectory from node i to node j with a bottleneck in space–time network (adopted from Lawson et al. [27])

Fig. 3 A sequence of pickup and drop-off links

time windows and vehicles' operating hour requirement, and more importantly, simplified queueing models (SQM) which can distinguish vehicles' speed in free-flow and congested conditions, under the assumption of constant bottleneck discharge capacities. The study by Lu et al. [26] provides a more systematic comparison among modes of point queues, spatial queues, and an extended version with time-dependent capacity and spillback along the backward wave. Along this line, one can extend the method from Lawson et al. [27] to calculate the spatial and temporal extents of queue and the actual waiting time spent on upstream of a bottleneck. The schematic trajectories of vehicle v from node i to node j in space–time network are shown in Fig. 2.

The detailed attributes of vehicles and passengers can be described in the coupled modules of simulation and optimization. Given a set of passengers P and their travel requests, pickup/delivery locations o_p/d_p for each passenger

p , a set of vehicles V with vehicle capacity Cap_v and other routing constraints, the simulation problem aims to find a feasible set of vehicle's space–time trajectories. Given an aggregated realization of traffic flow conditions from the simulation module, the optimization module needs to perform matching and provide system-optimal routing instructions between vehicles and passengers. In details, a space–time network can be built systematically using the following steps from Tong et al. [28]. As illustrated in Fig. 3, a vehicle could travel through a sequence of pickup and drop-off links for different passengers. For a passenger, his first pickup link along his trip is his own pickup link, while he can still travel through pickup/drop-off links of other passengers. This unique modeling feature is critically important for urban rail transit and public transportation bus systems (Table 2). A station should be represented as a pair of pickup and drop-off links, as well as a possible dedicated traveling-through link, as 3–5 and 6–8 in Fig. 3.

Table 2 Node sequences of passenger and vehicle in their paths (the pickup links for passengers are marked in bold)

Agent	Node sequence
V_1	depot ₁ - 1-2-3-4-5-6-7-8 -9-10-depot ₂ (1-2,4-5,7-8 are pickup links; 3-4,6-7,9-10 are drop-off links; 2-3,5-6,8-9 are moving links)
P_1	O ₁ - 1-2-3-4-5 -6-7-D ₁ (1-2 is his own pickup link; 6-7 is his own drop-off link; 2-3 and 5-6 are moving links, 4-5 is the others' pickup link)
P_2	O ₂ - 1-2 -3-4-D ₂ (1-2 is pickup link; 3-4 is drop-off link; 2-3 is moving link)
P_3	O ₃ - 7-8 -9-10-D ₃ (7-8 is pickup link; 9-10 is drop-off link; 8-9 is moving link)

Algorithm 1. Build a space-time network**Step 1:** Build space-time **vertex** set E

Add vertex (i, t) to E for $i \in N$ and each t

Step 2: Build space-time **transportation arc** set A^p/A^v for passengers/vehicles

Step 2.1: Add space-time **traveling** arc $(i, j, t, t + TT(i, j, t))$ to A^p/A^v , for link $(i, j) \in L$, where $TT(i, j, t)$ is the link travel time from node i to node j starting at time t

Step 2.2: Add a set of space-time **waiting** arcs for a pair of vertexes with the same node i , from time t to $t + 1$

Step 3: Add space-time **pick-up and drop-off arcs** for passengers to consider **request time window constraints**

Step 3.1: Add space-time pick up arc $(o_p', o_p, t, t + 1)$ to A^p at each time t ($EDT(p) \leq t < LAT(p)$) for each passenger p , where o_p' is o_p 's corresponding virtual node

Step 3.2: Add space-time drop off arc $(d_p, d_p', t, t + 1)$ to A^p at each time t ($EDT(p) < t \leq LAT(p)$) for each passenger p , where d_p' is d_p 's corresponding virtual node

Step 4: Add space-time **origin-departing and destination-returning arcs** for vehicles to consider **constraint of operating hours**

Step 3.1: Add space-time departing arc $(o_v', o_v, t, t + 1)$ to A^v at each time t ($EDT(v) \leq t < LAT(v)$) for each vehicle v , where o_v' is o_v 's corresponding origin node

Step 3.2: Add space-time arriving arc $(d_v, d_v', t, t + 1)$ to A^v at each time t ($EDT(v) < t \leq LAT(v)$) for each vehicle v , where d_v' is d_v 's corresponding destination node

3.2 Mathematic Formulations for Traffic and Public Transportation Optimization Problem on a Space-Time Network

The notations used in this section are listed in Tables 3, 4, and 5.

Focusing on analytically solvable equations, space-time network-based optimization used in our study needs to capture complex and important features including vehicle capacities, pickup, sharing policies, fare schemes, and

hailing strategies such as in-advance and on-demand requests. The optimization problem in DTALite-S can be formulated as a multi-commodity network flow model. The **objective function** shown in Eq. (1) is to minimize all vehicles and passengers' total costs.

$$\begin{aligned} \text{Min } Z = & \sum_{v \in V} \sum_{(i,j,t,t') \in A^v} C(v, i, j, t, t') \times y_{i,j,t,t'}(v) \\ & + \sum_{p \in P} \sum_{(i,j,t,t') \in A^p} C(p, i, j, t, t') \times y_{i,j,t,t'}(p) \end{aligned} \quad (1)$$

Table 3 Basic indices used to describe the space–time modeling framework

Index	Definition
M	Physical transportation network
G	Space–time transportation network
v	Vehicle index, $v \in V$
p	Passenger index, $p \in P$
$EDT(p)$	Earliest departure time of passenger p
$LAT(p)$	Latest arrival time of passenger p
$EDT(v)$	Earliest departure time of vehicle v
$LAT(v)$	Latest arrival time of vehicle v
o_v	Origin node of vehicle v , $o_v \in N$
o_p	Origin node of passenger p , $o_p \in N$
o'_p	Dummy node for passenger p 's origin
d_v	Destination node of vehicle v , $d_v \in N$
d_p	Destination node of passenger p , $d_p \in N$
d'_p	Dummy node for passenger p 's destination
(i, j, t, t')	Index of space–time traveling arc, $(i, j, t, t') \in A$
$TT(i, j, t)$	Link travel time from node i to node j starting at time t

Flow balance constraints for **physical path** at vehicle v 's node

$$\sum_{j \in N: (i, j) \in L} x_{ij}(v) - \sum_{j \in N: (j, i) \in L} x_{ji}(v) = \begin{cases} 1, & i = o_v \\ -1, & i = d_v \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

$$x_{ij}(v) \in \{0, 1\}, \forall i \in N, \forall v \in V$$

Flow balance constraint for **space–time path** at vehicle v 's vertex

$$\sum_{(j, t') \in E: (i, j, t, t') \in A^v} y_{ij, t, t'}(v) - \sum_{(j, t') \in E: (j, i, t', t) \in A^v} y_{ji, t', t}(v) = \begin{cases} 1, & (i, t) = (o_v, EDT(v)) \\ -1, & (i, t) = (d_v, LAT(v)) \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

$$y_{ij, t, t'}(v) \in \{0, 1\}, \forall (i, t) \in E, \forall v \in V$$

Flow balance constraints for **physical path** at passenger p 's node

$$\sum_{j \in N: (i, j) \in L} x_{ij}(p) - \sum_{j \in N: (j, i) \in L} x_{ji}(p) = \begin{cases} 1, & i = o_p \\ -1, & i = d_p \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

$$x_{ij}(p) \in \{0, 1\}, \forall i \in N, \forall p \in P$$

Flow balance constraint for **space–time path** at passenger p 's vertex

$$\sum_{(j, t') \in E: (i, j, t, t') \in A^p} y_{ij, t, t'}(p) - \sum_{(j, t') \in E: (j, i, t', t) \in A^p} y_{ji, t', t}(p) = \begin{cases} 1, & (i, t) = (o_p, EDT(p)) \\ -1, & (i, t) = (d_p, LAT(p)) \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

$$y_{ij, t, t'}(p) \in \{0, 1\}, \forall (i, t) \in E, \forall p \in P$$

Consistency constraints **between physical path and space–time path** for p

$$\sum_{(t, t') \in T} y_{ij, t, t'}(p) = x_{ij}(p), \forall p \in P, (i, j) \in L^p \quad (6)$$

Table 4 Basic parameters used to describe the space–time modeling framework

Notations	Definition
$TT(i, j, t)$	Link travel time from node i to node j starting at time t
Cap_v	Number of seats in vehicle v , e.g., 4 seats for passenger cars 20–30 seats for a bus, and 300 seats for urban rail train unit
$CapRoad_{i, j, t}$	Road capacity from node i to node j starting at time t , e.g., 1800 vehicles per hour per lane for freeway links, or precisely 1 per time interval for a track in urban rail transit, or multiple buses per time unit depending on the length of the bay in a bus station
$C(v, i, j, t, t')$	Vehicle-specific transportation cost on arc (i, j, t, t') traveled by vehicle v , including transportation costs, vehicle waiting time, converted through drivers' values of time
$C(p, i, j, t, t')$	Passenger-specific transportation cost on arc (i, j, t, t') traveled by passenger p , including transportation costs, passenger waiting time and converted through passengers' values of time

Table 5 Key variables used to describe the space–time modeling framework

Notations	Definition
$x_{ij}(p)$	Passenger routing variable (= 1, if physical arc (i, j) is selected by passenger p ; = 0, otherwise)
$x_{ij}(v)$	Vehicle routing variable (= 1, if physical arc (i, j) is selected by vehicle v ; = 0, otherwise)
$y_{ij, t, t'}(p)$	Passenger space–time routing variable (= 1, if space–time arc (i, j, t, t') is selected by passenger p ; = 0, otherwise)
$y_{ij, t, t'}(v)$	Vehicle space–time routing variable (= 1, if space–time arc (i, j, t, t') is selected by vehicle v ; = 0, otherwise)
$z(p, v)$	Passenger-to-vehicle marching variable (= 1, if passenger p is transported by vehicle v ; = 0, otherwise), $\forall p \in P, v \in V$

Fig. 4 Queue length and travel time various in simulation process, where WT is the waiting time

Consistency constraints **between physical path and space-time path** for v

$$\sum_{(t,t') \in T} y_{i,j,t,t'}(v) = x_{i,j}(v), \forall v \in V, (i,j) \in L^v \quad (7)$$

Road capacity constraints for all the vehicles using the same link

$$\sum_{v \in V} y_{i,j,t,t'}(v) \leq \text{CapRoad}_{i,j,t}, \forall (i,j) \in A^v, t \in T \quad (8)$$

Passenger carrying capacity constraints for vehicles

$$\sum_{p \in P} y_{i,j,t,t'}(p) \times z(p,v) \leq y_{i,j,t,t'}(v) \times \text{Cap}_v, \forall v \in V, (i,j,t,t') \in A^v \quad (9)$$

Matching constraints (10) ensures that each passenger is matched to exactly one vehicle

$$\sum_{v \in V} z(p,v) = 1, \quad \forall p \in P \quad (10)$$

Equations (2) to (5) impose passenger/vehicle flow balance constraints on both physical and space-time network. Equations (6) and (7) ensure that the path used in the physical network corresponds to the time-index trajectories in the space-time network for each agent. Both classes of MaaS vehicles and passengers use the offline scheduled routes from the passenger-to-vehicle assignment results in the MaaS optimization program. In the future, an on-line vehicle-to-passenger matching capability can be a nature extension; for example, following the line of research by Ma et al. [29] Alonso-Mora et al. [30], and Vazifteh et al. [31], constraint (8) can be viewed as a simplified version of point queue model with constant capacity in a space-time

network with waiting arcs, and the related modeling details can be found in Lu et al. [26]. Constraint (9) aims to satisfy the vehicle carrying capacity with combined Eq. (10), which ensures that each passenger is matched to exactly one vehicle. If we need to consider intermodal transfers in the public transportation problem, then one passenger may be served by multiple vehicles in a tour sequence and Eq. (10) needs to be further extended.

4 Simulation Process of Vehicular Loading and Passenger Pickup and Drop-off Services

4.1 Simulation Flowchart Based on Simple Data Structure

Illustrated in Algorithm 2, we need to perform two loops of time and agents across different links to check the available road and vehicle carrying capacity. As we follow a point queue-based system, without complicated data structure, we only need to concern about the key variables, namely arrival time and departure time of vehicle v on link l : $TA(v, l)$, $TD(v, l)$, as well as cumulative arrival/departure counts of vehicles on link l at time t , $A(l, t)$ and $D(l, t)$.

In step 3, illustrated in Fig. 4, the waiting time and queue length can be derived from the grant traffic state variables of $A(l, t)$ and $D(l, t)$. Overall, it could be memory consuming to store a full matrix of variables $TA(v, l)$, $TD(v, l)$, one can use dynamically allocated vectors to store and update the link sequences along their paths.

Algorithm 2. Simulation process in DTALite-S using a simplified queue based model**Step 1: Initialization: prepare input data with given agents' physical routes,**

A route could pass through a sequence of transportation, pick-up and drop-off links.

Step 2: Perform simulation

For ($t=0$; $t < T$; $t++$) // loop for each simulation time t

For ($l=0$; $l < L$; $l++$) // loop for each link in the network

Calculate link capacity $CapRoad_{i,j,t}$, based on cumulative arrival and departure counts $A(l, t-1)$, $D(l, t-1)$ at previous time interval $t-1$.

For ($v=0$; $v < V$; $v++$) // loop for each vehicle in current link l

If (vehicle v is ready to move on link l , i.e. $TD(v, l) = t$)

Check available capacity, if $CapRoad_{i,j,t} \geq 1$ then move vehicle v and the passengers carried in the vehicle, find the next link l' on its path, $TA(v, l') = t$, $TD(v, l') = t + FTTT(l')$ as the next available time to leave from link l' . Update capacity of link l by $CapRoad_{i,j,t} - PCE(v)$, update the cumulative flow counts $A(l, t)$, $D(l, t)$. $PCE(v)$ is the passenger car equivalent value of vehicle v .

For a **drop-off** link, if passenger and vehicle's next link is not the same, then the vehicle drops off the passenger and removes it from the current **passenger** list in the vehicle; and set $Cap_v = Cap_v + 1$.

For a **pick-up** link, the vehicle can pick up passengers from the link-based passenger waiting queue, if the vehicle's carrying capacity Cap_v is available, and further reduce Cap_v by $Cap_v - 1$.

Else

vehicle v needs to wait for next time interval, that is, $TD(v, l) = t+1$

Endif

Endfor// for loop each vehicle v

Endfor// for loop each link l

Endfor// for loop each time t

Step 3: Output data for statistics collection

As illustrated in Fig. 5, the software architecture designed in DTALite-S aims to bridge multi-agent simulation and optimization (integrating passenger-to-vehicle assignment and time-dependent routing) in the open-source DTALite-S modeling package. ODME (origin–destination matrix estimation) is considered throughout the dynamic assignment module adopted from Zhou et al. [32].

4.2 Further Discussions for Multimodal Environment

As a discretized space–time network-based simulation tool for the multimodal environment, we need to model various public transportation problems including routing and

scheduling of urban rail transit system, bus, taxi with ride-hailing strategies, and with in-advance and on-demand requests. Accordingly, Table 6 shows a detailed comparison of multi-mode agent representation and the related modeling focuses.

Finally, as a simplified or “student” version of DTALite on a fully discretized space–time network, DTALite-S has the following distinctions listed in Table 7.

Fig. 5 Schematic flowchart of major modeling components of DTALite-S

Table 6 Comparison of different applications in transportation network

Transportation mode	Agents	Focuses of modeling
Urban rail transit	<ul style="list-style-type: none"> • Passenger oversaturation • Passengers and vehicles have fixed routes 	<ul style="list-style-type: none"> • Waiting time • Walking link • Transfer for $x_{ij}(p)$ • Timetable
Bus	<ul style="list-style-type: none"> • $y_{ij}(v)$ is not stable/reliable, affect $y_{ij}(p)$ 	<ul style="list-style-type: none"> • Waiting time • Walking link • Transfer for $x_{ij}(p)$ • Bus service network design for $x_{ij}(v)$ • Timetable
Taxi	<ul style="list-style-type: none"> • Real-time passenger-to-vehicle matching 	<ul style="list-style-type: none"> • Affected by traffic congestion, capacity • Dynamically solve assignment of passengers and vehicles • Pickup and drop-off
Driving only	<ul style="list-style-type: none"> • Vehicles carry passengers 	<ul style="list-style-type: none"> • Traffic assignment problem with user equilibrium (UE) • Focus on $y_{ij}(v)$ and $x_{ij}(v)$, • Vehicle capacity and road capacity
Bike	<ul style="list-style-type: none"> • Passengers carry vehicles 	<ul style="list-style-type: none"> • Computational graph • Routing is decided by passenger

5 An ADMM-Based Framework to Understand Linkage Between Multi-agent-Based Optimization and Simulation

ADMM is a widely used optimization algorithm, and the literature of ADMM can be traced to classical papers by Douglas and Rachford [33] and Boyd et al. [20].

Essentially, it integrates problem decomposition techniques of augmented Lagrangian and block coordination descent. In our proposed framework, ADMM is firstly used to generate vehicle routing solutions for the multi-agent optimization problem. With Lagrangian multipliers and quadratic penalty terms, the complex public transportation mobility model can be decomposed into simple subproblems that can be solved by a sequential solution scheme.

Table 7 Major modeling enhancements from DTALite to DTALite-S

Index	DTALite	DTALite-S
Representation scale	Mesoscopic, link-based network	Range from mesoscopic link-based and microscopic cell-based network, with additional hyper space–time–state network details for vehicle routing problem with pickup and delivery
Traffic dynamics model	Point, spatial, and simplified kinematic wave-based queues	Point queue model
Agent type	Vehicles	Vehicles and passengers
Mode	Driving-only mode	Multiple modes
Simulation functionality	Traffic assignment, queue propagation	Traffic assignment, queue propagation, vehicle routing optimization
Algorithm	Label correcting algorithm, LR	Dynamic programming, LR, ADMM
Time-based simulation	Discrete-time in physical network with fixed/regular time interval (6 s)	Discrete-time in space–time network with flexible simulation time interval from 0.1 s to 1 min

The open-source vehicle routing package using the ADMM framework proposed by Yao et al. [34] can be downloaded from https://github.com/YaoYuBJTU/ADMM_Python.

Consider a space–time network problem with two agents v_1 and v_2 . Given v_2 's trajectory up to time $t + h$, we need to schedule trajectory of v_1 from t to $t + h$, where h is the optimization time horizon in the future. We can interpret the above problem from perspectives of simulation and optimization.

$$\begin{aligned} \text{Min } & \sum_{v \in V} \sum_{(i,j) \in A} C_{ij}(v) \times y_{ij}(v) \\ \text{s.t. } & \sum_{v \in V} \sum_{(i,j) \in A^v} y_{ij}(v) \leq 1 \end{aligned} \quad (11)$$

The standard **Lagrangian relaxation** form for problem (11) can be rewritten as Eq. (12), where $\lambda_{(i,j)}$ is the vector of Lagrangian multiplier for different links.

$$\begin{aligned} L(v, \lambda_{ij}) = & \sum_{v \in V} \sum_{(i,j) \in A} C_{ij}(v) \times y_{ij}(v) \\ & + \sum_{(i,j) \in A^v} \lambda_{(i,j)} \times \left(\sum_{v \in V} \sum_{(i,j) \in A^v} y_{ij}(v) - 1 \right)^+ \end{aligned} \quad (12)$$

The **ADMM function** can be written as Eq. (13), and the algorithm will apply a block coordinate decomposition scheme for each agent v .

$$\begin{aligned} L(v, \lambda_p, \rho) = & \sum_{v \in V} \sum_{(i,j) \in A} C_{ij}(v) \times y_{ij}(v) \\ & + \sum_{(i,j) \in A^v} \lambda_{(i,j)} \times \left(\sum_{v \in V} \sum_{(i,j) \in A^v} y_{ij}(v) - 1 \right)^+ \\ & + \left(\frac{\rho}{2} \right) \left\| \left(\sum_{v \in V} \sum_{(i,j) \in A^v} y_{ij}(v) - 1 \right)^+ \right\|_2^2 \end{aligned} \quad (13)$$

Through the record of $CapRoad_{i,j,t}$ in Algorithm 2, the simulation mechanism in DTALite-S first removes the infeasible space–time regions occupied by vehicle v_2 , to ensure the full feasibility trajectories of v_1 . Recall that, $CapRoad_{i,j,t} = 0$ indicates that the space–time resource capacity at link (i, j) at time t has been used by one previously scheduled vehicle, otherwise.

As shown in Table 8, we want to show how the sequential ADMM decomposition method can be linked to the vehicle loading process inside simulation. In this context, whether a vehicle can use the resource at (i, j, t) depends on the value of penalty parameters ρ in Eq. (13). Of course, an infinite value of ρ can always lead to feasible solutions, which is equivalent to setting $CapRoad_{i,j,t} = 0$ by removing infeasible space–time solution regions. On the other hand, a small value of ρ could still allow different vehicles to compete for the limited road resources, through an iterative process, depending on their own priority and waits on the objective functions. From the ADMM

Table 8 Conceptual algorithmic comparison between ADMM with simulation processes in the space–time network representation

ADMM as iterative optimization procedure	One-shot simulation
For ($k=0; k < K; k++$) // k is optimization iteration	For ($t=0; t < T; t++$) // loop for each time t
For ($v=0; v < V; v++$) // loop for each vehicle in the network	For ($v=0; v < V; v++$) // loop for each vehicle v in the network
For ($t=0; t < T; t++$) // loop for each time t	Given previously scheduled trajectories and already used road recourse at different space time slots (i, t)
Update the cost associated with penalty term ρ	Schedule feasible trajectory from time t to $t+h$ for each vehicle v
Find space-time shortest path for each vehicle v with arc cost $C_{i,j}(v)$, feasibility penalty and Lagrangian multipliers $\lambda_{(i,j)}$	Mark infeasibility for newly scheduled trajectories
Endfor //for loop each t	Endfor // for loop each vehicle v
Endfor // for loop each vehicle v	Endfor // for loop each time t
Update Lagrangian multipliers $\lambda_{(i,j)}$	
Update penalty term ρ , typically from small to large	
Compute up bound and lower bound	
When reach termination condition	
Endfor // for each iteration k	

modeling perspective, in each iteration k , the simulation process can be viewed as a one-shot procedure from time 0 to time T , with the infinite penalty parameters to ensure the full feasibility of space–time agent trajectories.

Essentially, the ADMM-based iterative optimization procedure can optimize the system-wide costs by iteratively scheduling individual vehicle trajectories with a relatively small ρ and then increasing the value of ρ further aims to enforce the primal feasibility. In contract, the simulation-based process makes scheduling decisions only based on purely local information at current time and only schedules the trajectories at a short time horizon. Interested readers can further study the use of ADMM in a distributed multi-agent optimization environment, e.g., for distributed automated vehicles, based on a recent study by Nedic and Ozdaglar [4].

6 Case Study

In this section, we implement the DTALite-S engine to simulate the passenger traveling process in the Beijing subway network as shown in Fig. 6, which contains of 18 urban rail transit lines, 343 stations, and 52 transfer stations.

In the proposed modeling framework, Beijing subway network is represented by a physical network with 2401 nodes and 3510 links. A total of 6699 train services are modeled into the DTALite-S as vehicle schedules. Up to 1,393,093 passenger transaction record samples during 7:00 to 9:00 on a typical workday are obtained from the AFC system in the Beijing subway network as input passenger agents. To implement Beijing subway case, the total computing time is about 1560 s. For illustration purpose, we choose four time intervals during the morning peak hours in Fig. 7 to show the passenger crowding levels over the whole Beijing subway network.

We can also obtain the time-dependent passenger waiting queues at platforms. Figure 8 shows passenger queue lengths and crowding levels in both up and down directions of Yizhuang line in Beijing.

To assess a more complete set of simulation performances, we also calculate the absolute deviation value between the simulated arrival time and actual arrival time of each passenger, as shown in Fig. 9. Specifically, the absolute deviations of 65% passengers are within 3 min and the absolute deviations of 82% passengers are within 5 min.

Fig. 6 Layout of the Beijing subway network

Fig. 7 Passenger crowding in trains over the network

Fig. 8 Passenger waiting queues at platforms and crowding levels in trains on Yizhuang line

7 Conclusions

Optimization tends to solve tactical/operational issues, while simulation aims to resolve more complex and realistic transportation problems. A number of researchers have devoted a great amount of efforts to integrating both optimization approaches and simulation tools to pursue future advanced mobility solutions. For example, stochastic programming and approximate dynamic programming often use simulation to estimate expected values within an optimization framework.

The proposed discrete space–time network-based modeling framework can be used to evaluate and further

optimize transport strategies from new demand and supply perspectives. It intends to bridge the modeling gaps for solving complex transportation problems, such as multi-modal transportation network design, VRP, urban rail transit scheduling, bus operating, etc. With the embedded Lagrangian relaxation and ADMM algorithms, the framework is able to shed more lights on many fundamental research issues in large-scale dynamic traffic assignment, mesoscopic traffic simulation, and vehicle route optimization. The proposed modeling framework is used to simulate the passenger traveling process in Beijing subway network. Via the ADMM-based solution approach, queue lengths at platforms, in-vehicle congestion levels and

Fig. 9 The absolute deviation between the simulated arrival times and actual arrival times of all passengers

absolute deviation of travel times are obtained within 1560 seconds. The numerical examples indicate that the proposed simulation engine DTALite-S is able to efficiently simulate passenger traveling process.

In our future research, a more comprehensive and microscopic cell-based simulation framework will be further studied to simulate more detailed travel decisions, such as passenger route choice lane changing behavior, and car flowing behavior. In addition, we will apply and extend the proposed open-source offline modeling framework to many emerging transportation applications, in a real-time and distributed computing environment.

Acknowledgements This research project, especially the large-scale Beijing Subway network and smart card data set, has been supported through Beijing Key Laboratory of Urban Traffic Operation Simulation and Decision Support and Beijing International Science and Technology Cooperation Base of Urban Transport. The last author is partially funded by National Science Foundation, USA, under NSF Grant No. CMMI 1538105 “Collaborative Research: Improving Spatial Observability of Dynamic Traffic Systems through Active Mobile Sensor Networks and Crowdsourced Data” and NSF Grant No. CMMI 1663657. “Real-time Management of Large Fleets of Self-Driving Vehicles Using Virtual Cyber Tracks”.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Osorio C, Bierlaire M (2013) A simulation-based optimization framework for urban transportation problems. *Oper Res* 61(6):1333–1345
- Xiong C, Zhu Z, Chen X, Zhang L (2018) Optimal travel information provision strategies: an agent-based approach under uncertainty. *Transp B Transp Dyn* 6(2):129–150
- Mahmassani HS, Fei X, Eisenman S, Zhou X, Qin X (2005) DYNASmart-X evaluation for real-time TMC application: CHART test bed. Maryland Transportation Initiative, University of Maryland, College Park, Maryland, pp 1–144
- Nedic A, Ozdaglar A (2009) Distributed subgradient methods for multi-agent optimization. *IEEE Trans Autom Control* 54(1):48–61
- Fagnant D, Hall CJ, Kockelman KM (2014) The travel and environmental implications of shared autonomous vehicles, using agent-based model scenarios. *Transp Res Part C* 40(1):1–13
- Newell GF (1993) A simplified theory of kinematic waves in highway traffic, part iii: multi-destination flows. *Transp Res Part B Methodol* 27(4):281–287
- Zhou X, Taylor J, Pratico F (2014) DTALite: a queue-based mesoscopic traffic simulator for fast model evaluation and calibration. *Cogent Eng* 1(1):961345
- Spieser K, Treleaven KB, 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. Springer International Publishing, Road Vehicle Automation
- Yu Y, El Kamel A, Gong G, Li F (2014) Multi-agent based modeling and simulation of microscopic traffic in virtual reality system. *Simul Model Pract Theory* 45:62–79
- Bierlaire M (2015) Simulation and optimization: a short review. *Transp Res Part C Emerg Technol* 55:4–13
- Kloostera B, Roorda MJ (2017) Fully autonomous vehicles: analyzing transportation network performance and operating scenarios in the Greater Toronto Area, Canada. In: TRB 2017 annual meeting
- Khattak A, Yangsheng J, Lu H, Juanxiu Z (2016) Modeling and simulation of metro transit station walkway as a state-dependent queuing system based on the phase-type distribution. *Int J Traffic Transp Eng* 5(5):103–111
- Liang X, de Almeida Correia GH, Van Arem B (2016) Optimizing the service area and trip selection of an electric automated taxi system used for the last mile of train trips. *Transp Res Part E Logist Transp Rev* 93:115–129
- Mahmassani HS (2016) 50th anniversary invited article—autonomous vehicles and connected vehicle systems: flow and operations considerations. *Transp Sci* 50(4):1140–1162

15. Qu Y, Zhou X (2017) Large-scale dynamic transportation network simulation: a space-time-event parallel computing approach. *Transp Res Part C Emerg Technol* 75:1–16
16. Martinez LM, Correia GH, Viegas JM (2015) An agent-based simulation model to assess the impacts of introducing a shared-taxi system: an application to Lisbon (Portugal). *J Adv Transp* 49(3):475–495
17. Golubev K, Zagarskikh A, Karsakov A (2018) A framework for a multi-agent traffic simulation using combined behavioral models. *Procedia Comput Sci* 136:443–452
18. Sun Y, Zhang C, Dong K, Lang M (2018) Multiagent modelling and simulation of a physical internet enabled rail-road intermodal transport system. *Urban Rail Transit* 4(3):141–154
19. Wen J, Chen YX, Nassir N, Zhao J (2018) Transit-oriented autonomous vehicle operation with integrated demand-supply interaction. *Transp Res Part C Emerg Technol* 97:216–234
20. Boyd S, Parikh N, Chu E, Peleato B, Eckstein J (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found Trends Mach Learn* 3(1):1–122
21. Mahmoudi M, Zhou X (2016) Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: a dynamic programming approach based on state-space-time network representations. *Transp Res Part B Methodol* 89:19–42
22. Tong LC, Zhou L, Liu J, Zhou X (2017) Customized bus service design for jointly optimizing passenger-to-vehicle assignment and vehicle routing. *Transp Res Part C Emerg Technol* 85:451–475
23. Wei Y, Avci C, Liu J, Belezamo B, Aydın N, Li PT, Zhou X (2017) Dynamic programming-based multi-vehicle longitudinal trajectory optimization with simplified car following models. *Transp Res Part B Methodol* 106:102–129
24. Zhou X, Tong L, Mahmoudi M, Zhuge L, Yao Y, Zhang Y, Shang P, Liu J, Shi T (2018) Open-source VRPLite package for vehicle routing with pickup and delivery: a path finding engine for scheduled transportation systems. *Urban Rail Transit* 4(2):68–85
25. Zhao M, Li X, Yin J, Cui J, Yang L, An S (2018) An integrated framework for electric vehicle rebalancing and staff relocation in one-way carsharing systems: model formulation and Lagrangian relaxation-based solution approach. *Transp Res Part B Methodol* 117:542–572
26. Lu CC, Liu J, Qu Y, Peeta S, Roupail NM, Zhou X (2016) Ecosystem optimal time-dependent flow assignment in a congested network. *Transp Res Part B Methodol* 94:217–239
27. Lawson TW, Lovell DJ, Daganzo CF (1997) Using input-output diagram to determine spatial and temporal extents of a queue upstream of a bottleneck. *Transp Res Rec* 1572(1):140–147
28. Tong L, Zhou X, Miller HJ (2015) Transportation network design for maximizing space-time accessibility. *Transp Res Part B Methodol* 81:555–576
29. Ma J, Li X, Zhou F, Hao W (2017) Designing optimal autonomous vehicle sharing and reservation systems: a linear programming approach. *Transp Res Part C Emerg Technol* 84:124–141
30. Alonso-Mora J, Samaranayake S, Wallar A, Frazzoli E, Rus D (2017) On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proc Natl Acad Sci* 114(3):462–467
31. Vazifteh MM, Santi P, Resta G, Strogatz SH, Ratti C (2018) Addressing the minimum fleet problem in on-demand urban mobility. *Nature* 557(7706):534
32. Lu CC, Zhou X, Zhang K (2013) Dynamic origin-destination demand flow estimation under congested traffic conditions. *Transp Res Part C Emerg Technol* 34:16–37
33. Douglas J, Rachford HH (1956) On the numerical solution of heat conduction problems in two and three space variables. *Trans Am Math Soc* 82(2):421–439
34. Yao Y, Zhu X, Dong H, Wu S, Wu H, Zhou X (2018). An alternating direction method of multiplier based problem decomposition scheme for iteratively improving primal and dual solution quality in vehicle routing problem. Working paper