**Open-source** **public transportation mobility simulation engine DTALite-S: a discretized space-time network based modeling framework for** **bridging multi-agent simulation and optimization**

**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 multi-modal 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 ADMM (Alternating Direction Method of Multipliers) 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.

**Key words**

Space-time network, dynamic traffic assignment, multi-agent simulation, Lagrangian relaxation, ADMM (Alternating Direction Method of Multipliers)

# Introduction

# Simulation process of vehicular loading and passenger pick-up 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 be concerned about very few key variables namely arrival time and departure time of vehicle  on link: *, ,* as well as cumulative arrival/ departure counts of vehicles on link at time , ) and ).

**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** (=0;<; ++) // loop for each simulation time |
| **For** (=0; <; ++) // loop for each link in the network |
| Calculate link capacity , based on cumulative arrival and departure counts ), ) at previous time interval . |
| **For** (=0;<; ++) // loop for each vehicle in current link |
| **If** (vehicle is ready to move on link *l*, i.e. ) |
| Check available capacity, if then move vehicle *v* and the passengers carried in the vehicle, find the next link on its path, , as the next available time to leave from link Update capacity of link by , update the cumulative flow counts ), ). is the passenger car equivalent value of vehicle . |
| 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 .  For a **pick-up** link, the vehicle can pick up passengers from the link-based passenger waiting queue, if the vehicle’s carrying capacity is available, and further reduce by |
| **Else**  vehicle *v* needs to wait for next time interval, that is, +1  **Endif** |
| **End**// for loop each vehicle |
| **End**// for loop each link |
| **End**// for loop each time |
| **Step 3: Output data for statistics collection** |

In step 3, Illustrated in Fig. 4, the waiting time and queue length can be derived from the grant traffic state variables of ) and ). Overall, it could be memory consuming to store a full matrix of variables *,* , one can use dynamically allocated vectors to store and update the link sequences along their paths.



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

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, time-dependent routing) in the open-source DTALite-S modeling package.



**Fig. 5** Schematic flow chat of major modeling components of DTALite-S