**Open-source** **public transportation mobility simulation engine DTALite-S: a discrete 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, DTALite-S is further designed and implemented 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 the simulation problem on a space-time event 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 LR (Lagrangian relaxation) method and a time-oriented sequential simulation procedure to handle many general applications. A number of illustrated examples are presented to demonstrate the applicability of the proposed simulation framework under various transportation mobility scenarios.

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

# Introduction

## Literature review

There are numbers of excellent reviews on multiple agents traffic assignment and traffic simulation.

(Zhou and Taylor, 2014) described the mesoscopic traffic simulation used the Newell’s kinematic approach (Newell, 1993) to simulate traffic movement on large-scale networks and to simulate large scale networks with millions of agents. (Mahmassani et al., 2004) use flow-density relationships to predict time-dependent traffic flows in DYNASMART. (Kirill Golubev et al. 2018) presented a novel agent-based traffic modelling framework, which can allow the user to set a specific model for each supported class. (Yue Yu et al. 2014) focuses on the modeling and simulation of microscopic traffic behavior in virtual reality system using multi-agent technology, a hierarchical modular modeling methodology and distributed simulation. (Igor TCHAPPI HAMAN et al. 2017) used a multilevel approach to support dynamic selection of the level during the simulation. (Arnaud Doniec et al. 2008) propose a multi-agent behavioral model based on the opportunistic individual behaviors to deal road traffic simulation. (Mahmassani, 2016) proposed a microsimulation framework featuring varying behavioral mechanisms for the three classes of vehicles, which was used to examine the throughput and stability questions through a series of experiments under varying market penetration rates of autonomous and/or connected vehicles. (Yan Sun et al., 2018) presents an agent-based simulation that could be used in urban rail transit systems serving urban distribution and passenger transportation. Based on kinematic wave model, (Yunchao Qu et al., 2017) present a computationally efficient parallel-computing framework for traffic simulation. (Wei et al., 2017) described a simplified multi-vehicle trajectory optimization based on a set of integer programming and dynamic programming models.

(Meng and Zhou, 2014) developed a computationally efficient algorithm, based on Lagrangian relaxation algorithms and time-dependent shortest path algorithms, for routing and timing a large number of trains on a medium size network. (Wei et al. 2015) developed efficient DP algorithms to guarantee the minimal safety distance between vehicles by a shifted state representation during searching the optimal trajectories. (Tong and Zhou, 2017) based on the Lagrangian decomposition and a space-time prism, developed a joint optimization model for public transportation services.

The literature of ADMM can be traced by the classical paper (Glowinski, 1975), and many researches based on it in the field of convex programming. (Angelia Nedic et al., 2009) distributed computation model for optimizing a sum of convex objective functions corresponding to multiple agents. (Stephen Boyd et al. 2010) discuss general distributed optimization, extension to the nonconvex setting and efficient implementation. (Zhou et al., 2017) describes the space-time-state modeling process of VRPPDTW using a hyper network representation. And embedded in a column generation or Lagrangian relaxation framework to handle many general applications. (Mahmoudi and Zhou，2016) based on state-space-time network present a time-dependent DP framework for single vehicle VRPPDTW problems.

## Motivation

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

Among widely used traffic simulation tools, DTALite is a queue-based mesoscopic traffic simulator, documented in the paper by (Zhou and Taylor, 2014). It is an open-source mesoscopic DTA (dynamic traffic assignment) simulation package, in conjunction with the Network explorer for Traffic Analysis (NeXTA) graphic user interface, developed 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. In this research, we mainly focus on how to provide a modeling framework to bridge simulation and optimization, and to potentially address different analytical and practical questions.

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.

(2) Offer a unified framework with pick up and drop off events that cover different traveling activities from driving-only to multiple modes, across a wide range of 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 combined 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).

## Modeling enhancements from DTALite to DTALite-S

# Problem statement and modeling framework

Given 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 need to perform **(1) space-time network-based traffic assignment** to find the shortest path for each agent, particularly the background vehicles, **(2) transportation mobility optimization**, which assign the tasks of serving passengers to specific vehicles in a cost-optimal way, and **(3) traffic simulation** for calculating traffic states in the transportation network. The system output **(4)** includes **time-dependent travel times**, and vehicle and passenger **space-time trajectories.**

Add questions

With new mobility modes and solutions—MaaS, developing, we need integrated public transit to reduce congestion, improve mobility and reduce pollution. Some substantive questions need to be addressed include: (a) How could we consider system-level cost and benefits through integrated public transit to implement system optimization ? (b) How will we used agent-based simulation tools to improve new mobility modes? (c) How to improve system flexibility and accessibility with multitude of modes? The major research objectives in this paper is to develop framework to improve public transportation mobility.

## Discretized space-time network construction

Given the input data, the physical transportation network is expanded into a two dimensional space-time network. Beside typical space-time traveling/waiting arcs, additional space-time arcs, such as pick up and drop off space-time arc are also be constructed for modeling the vehicle routing process. More precisely, consider a physical transportation network  with a finite set of nodes and a finite set of links where nodes  and directed link . A space-time network , where is set of space-time vertexes, is space-time link, can be constructed for transportation network under planning time horizon . Each arc indicates a directed space-time path from node departing at time to node arriving at time . Because of the space-time network structure, it is easy to model passengers’ travel requests, vehicles’ travel times changing over time. Agent from node to node in space-time network is shown in Fig. 4.



**Fig. 4** **Agent from node to node in space-time network**

The detailed attributes of vehicles and passengers can be described further. Given a set of passengers and their travel requests, as pickup/delivery locations for each passenger , a set of vehicles with vehicle capacity and other routing constraints, the simulation problem aims to find a feasible set of vehicle’s path and timetables for each vehicle in the vehicle set under certain traffic conditions. The optimization module needs to perform, if needed, matching between vehicles and passengers. In details, a space-time network can be built using the following steps (Tong et al. 2015).

**Algorithm 1.** Build a space-time network

|  |
| --- |
| **Step 1: Build space-time vertex set** |
| Add vertex to for and each |
| **Step 2: Build space-time arc set for passengers/vehicles** |
| Step 2.1: Add space-time traveling arc to , for link , where is the link travel time from node to node starting at time |
| Step 2.2: Add a set of space-time waiting arcs for a pair of vertexes with the same node , from time to  **Step 3: Add space-time pick up and drop off arcs**  Step 3.1: Add space-time pick up arc to at each time () for each passenger , where is corresponding virtual node  Step 3.2: Add space-time drop off arc to at each time () for each passenger , where is ’s corresponding virtual node |

The notations used in this section are listed in Table 2, Table 3 and Table 4.

**Table 2 Basic indices used to describe the space-time simulation framework**

|  |  |
| --- | --- |
| **Index** | **Definition** |
|  | Physical transportation network |
|  | Space-time transportation network |
|  | Vehicle index, *v* |
|  | Passenger index, *p* |
|  | Earliest departure time of passenger |
|  | Latest arrival time of passenger |
|  | Earliest departure time of vehicle |
|  | Latest arrival time of vehicle |
|  | Origin node of vehicle , |
|  | Origin node of passenger , |
|  | Dummy node for passenger origin |
|  | Destination node of vehicle , |
|  | Destination node of passenger , |
|  | Dummy node for passenger destination |
|  | Index of space-time traveling arc, |
|  | Link travel time from node to node starting at time |

**Table 3 Basic parameters used to describe the space-time simulation framework**

|  |  |
| --- | --- |
| **Notations** | **Definition** |
|  | Link travel time from node to node starting at time |
|  | Number of seats in vehicle , e.g., 4 seats for passenger cars  20-30 seats for a bus, and 300 seats for urban rail train unit. |
|  | Road capacity from node to node starting at time, e.g., 1800 vehicles per hour per lane |
|  | Transportation cost on arc traveled by vehicle , including transportation costs, passenger waiting time and vehicle waiting time, converted through the values of time |
|  | Transportation cost on arc traveled by passenger , including transportation costs, passenger waiting time and vehicle waiting time, converted through the values of time |

**Table 4 Core variables used to describe the space-time optimization problem**

|  |  |
| --- | --- |
| **Notations** | **Definition** |
|  | Vehicle routing variable (=1, if physical arc () is selected by passenger ; =0, otherwise) |
|  | Vehicle routing variable (=1, if physical arc () is selected by vehicle; =0, otherwise) |
|  | Vehicle space-time routing variable (=1, if space-time arc () is selected by passenger =0, otherwise) |
|  | Vehicle space-time routing variable (=1, if space-time arc () is selected by vehicle ; =0, otherwise) |
|  | Passenger-to-vehicle marching variable (=1, if passenger is transported by vehicle ; =0, otherwise), |

## Vehicle routing models as general mobility modeling building block in the space-time network

The optimization problem in DTALite-S can be formulated as a multi-commodity network flow model with a number of special capacity and assignment constraints. The goal is to minimize the total cost of passengers and vehicles, in Eq. (1) subjects to constraints (2)-(10).

The **objective function** is to minimize the vehicle  and the passenger total **space-time path** cost:

|  |  |
| --- | --- |
|  | (1) |

Flow balance constraints for **physical path** at vehicle node:

|  |  |
| --- | --- |
|  | (2) |

Flow balance constraint for **space-time path** at vehicle vertex:

|  |  |
| --- | --- |
|  | (3) |

Flow balance constraints for **physical path** at passenger *p* node:

|  |  |
| --- | --- |
|  | (4) |

Flow balance constraint for **space-time path** at vehicle vertex:

|  |  |
| --- | --- |
|  | (5) |

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

|  |  |
| --- | --- |
|  | (6) |

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

|  |  |
| --- | --- |
|  | (7) |

**Road capacity** constraints for vehicles in **space-time network**:

|  |  |
| --- | --- |
|  | (8) |

**Passenger** carrying **capacity** constraints for vehicles:

|  |  |
| --- | --- |
|  | (9) |

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

|  |  |
| --- | --- |
|  | (10) |

Add Discussion

The objective function of proposed model is to minimize the total costs of passengers and vehicles. Equations (2)-(5) by constitute feasible paths from the origin to the destination to guarantees flow balance for passengers and vehicles respectively in physical network and space-time network. Equations (6) and (7) synthesis that physical network correspond to space-time network for each agent. Constraints (8) and (9) are used to satisfied the capacity with combined equation (10), which ensures that each passenger is matched to exactly one vehicle. If the public transportation problem with transfers, one passenger may be served by multi-vehicles, then the above equation (10) need to be further extended.

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. 3 illustrates the model structure of DTALite-S. Fig.4 is the data flow chart of DTALite-S.

Relation of variable blocks in simulator



**Fig. 3 Software architecture with major modeling components of DTALite-S**

## Simulation process of point queue-based network vehicular loading and passenger pickup and dropoff services

The notations of the detailed simulation process used in this paper are listed in Table 4. Illustrated in Algorithm 1. We need to perform multiple loops of time and agents across different links to calculate the travel time, waiting time and the queue length.

**Algorithm 2.** Detail simulation process in DTALite-S

|  |
| --- |
| **Step 1: Initialization: prepare input data** |
| **Step 2: Perform simulation** |
| For (=0;<; ++) // loop for each simulation time |
| For (=0; <; ++) // loop for each link in the network |
| Calculate link capacity |
| For (=0;<; ++) // loop for each vehicle in current link *l* |
| **If** (vehicle is ready to move on link exit queue ) |
| check capacity and move according to Eq. (12) and (13) for vehicles and the passengers carried in the vehicle |
| execution actions according to the service type of link ; if passenger’s next link is equal to vehicle’s, execution moving; if passenger’s next link is not equal to vehicle’s, execution pick up or drop off according to service type |
| update link capacity and vehicle and, according to Eq. (11) |
| **Else** wait for next time interval |
| End for loop each vehicle |
| End for loop each link |
| End for loop each time |
| **Step 3: Output data for** statistics collection |

**Table 4** **Intermediate variables used in simulation framework, which can be derived from the core variables**

|  |  |
| --- | --- |
| **Parameters** | **Definition** |
|  | Arrival time of vehicle in its link with origin node |
|  | Departure time of vehicle in its link with destination node |
|  | Vehicle link travel time from node to node starting at time |
|  | Vehicle link free flow travel time from node to node starting at time |
|  | Vehicle waiting time from node to node and in node |
| ) | Cumulative arrival counts of vehicles from node to node at time |
| ) | Cumulative departure counts of vehicles from node to node at time |
|  | Queue length of vehicle in node at time |
|  | Cumulative arrival counts of vehicles from node to node at little interval , . |
|  | Cumulative departure counts of vehicles from node to node at little interval , . |



**Fig. 6 Queue length and travel time various in simulation process**



(Which one is better?)

In Fig.6 we can see more details about agents in simulation process form Fig.5We can’t obtain the results of intermediate simulation, but we can use build the model to recur the process. The function is started as expression (11) - (14).

Vehicle time changes in its link along that path in space-time network:

|  |  |
| --- | --- |
|  | (11) |

The following equations are needed for the statistics collection module/step 3

Cumulative arrival counts of vehicles in node to node at time :

|  |  |
| --- | --- |
|  | (12) |

Cumulative departure counts of vehicles in node to node at time :

|  |  |
| --- | --- |
|  | (13) |

Queue length of vehicle in node to node at time :

|  |  |
| --- | --- |
|  | (14) |

# An ADMM based theoretical framework to understand the linkage between multi-agent based optimization and simulation

ADMM is not a new algorithm, it is just integrated many classic optimization algorithm, and then combines the problems encountered in modern society to propose a more general, better implementation of distributed computing framework. It is suitable for optimizing the sum of objective functions of multiple agents, which are convex but not necessarily smooth.

Consider a space-time network problem with two agents and in the DTALite-S. Given trajectory up to time , schedule trajectory from to , where is the optimization time horizon in the future and is infinite that always feasible solution by removing infeasible space time solution regions. The simulation mechanism will first remove the infeasible space time regions occupied by vehicle , to ensure the full feasibility of trajectories. Vertex If vertex , indicates that the space time resource at node at time has been used by one previously scheduled vehicle. If vertex , indicate that no more vehicles can schedule to use this space time road/or transit node resources. In the context of ADMM modeling framework, whether can use the resource at is depend on. The infinite that always feasible solution by removing infeasible space time solution regions. The ADMM models used in space-time network is started as expression (21) to (28).

**Algorithm 3.** Build a formulation of ADMM in space-time network

|  |
| --- |
| **Step 1: Initialization** |
| **Step 2: Perform simulation** |
| For (=0;<; ++) // loop for each simulation time |
| For (=0;<; ++) // loop for each vehicle in the network |
| Given previously scheduled trajectories and already used road recourse at different space time slots |
| Schedule feasible trajectory from time to for vehicle |
| Mark infeasibility for newly scheduled trajectories |
| End for loop each vehicle |
| End for loop each time |

**Table 4 Basic parameters used in ADMM model to describe the space-time network**

|  |  |
| --- | --- |
| **Parameters** | **Definition** |
|  | Vehicle index, |
|  | Planning time horizon， |
|  | Number of seats in vehicle , e.g., 4 seats for passenger cars,  20-30 seats for a bus, and 300 seats for urban rail train unit. |
|  | multipliers |
|  | Given penalty coefficient |
|  | iterations |

The primal problem Eq. (1) simplifier expressed as (15).(我想写两个车，哪个对)

|  |  |
| --- | --- |
|  | (15) |

The main constraints from Eq. (8) is easy constrain expressed as (16)

|  |  |
| --- | --- |
|  | (16) |

The main constraints from Eq. (10) is hard constrain expressed as (17)

|  |  |
| --- | --- |
|  | (17) |

The **Lagrangian relaxation** form for problem (15) can be written as Eq. (18).

|  |  |
| --- | --- |
|  | (18) |

The **ADMM objection function** can be written as Eq. (19).

|  |  |
| --- | --- |
|  | (19) |

The **ADMM** variables and multipliers are updated separately and sequentially **constrains** can be written as Eq. (20)-(23).

**Vehicle optimization**, with given vehicle space-time trajectory

|  |  |
| --- | --- |
|  | (20) |

**Vehicle optimization**, with given vehicle space-time trajectory.

|  |  |
| --- | --- |
|  | (21) |

**Lagrangian relaxation multiplier updating** for a given penalty term .

|  |  |
| --- | --- |
|  | (22) |
|  | (27) |

Simple examples show that ADMM is slow to converge to high accuracy. However, it is practically useful in most cases when modest accuracy is sufficient. And can be increased to speed-up the convergence if necessary.



Fig.7 the difference iteration process of ADMM and simulation

Compared with the standard augmented Lagrangian method, the distinctive feature of ADMM is that the problem is decomposed into a specific block structure. In this sense, the large-scale problem can be decomposed into several sub-problems then solved in an alternating pattern. In other words, the ADMM approach can be viewed as robust dual decomposition or decomposable method of multipliers. Simulation keep full primal feasibility, and dual feasibility is in fact omitted.

On the other hand, ADMM-based iterative optimization procedure to reduce the costs by iteratively scheduling individual vehicle trajectories with a relatively small . Then increasing the value of further aims to enforce the primal feasibility.

In contract, the simulation based process described above, based on purely local information at current time, only schedules the trajectories at one iteration, across all time intervals t and across all vehicles v,

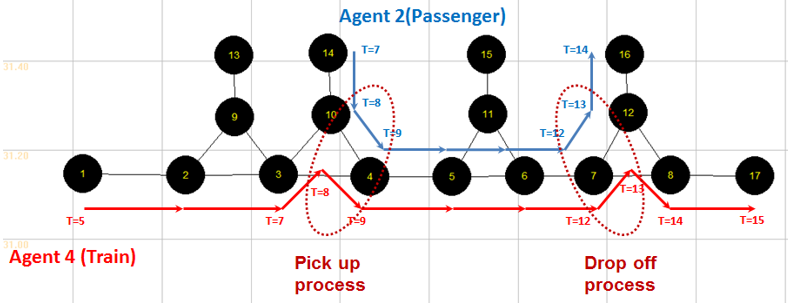
# Case study

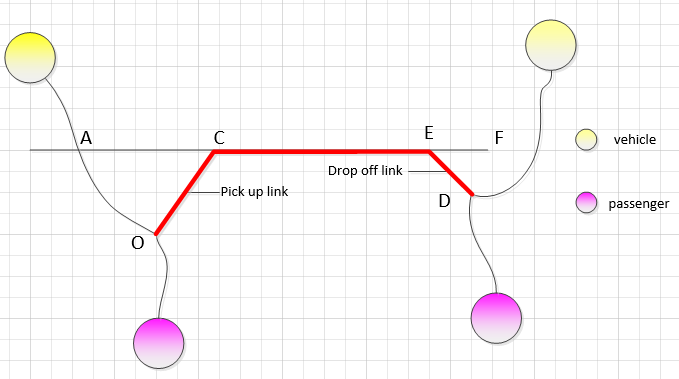
Discussion

DTALite-S is a discrete space-time network based modeling framework and it can solve various public transportation problems. The agents in urban rail transit involved passenger and vehicle, and sometimes, the passengers are oversaturation that we need consider station capacity. The other features in urban rail transit is passenger and vehicle has fixed routing. Agents in bus operations involved passenger and vehicle, and sometimes the vehicles can affect the passenger’s behavior. Accidental, bus timetable is not punctual, sometime it will be late because of traffic congestion or traffic accident. In Table 5 show the detail application comparison in transportation network and the main problem need to be solved.

**Table 5 Application comparison in transportation network**

|  |  |  |
| --- | --- | --- |
| **Mode in transportation** | **Agent and z(p,v) relationship** | **Focus of Problem** |
| Urban rail transit | P sometimes are oversaturation;  P and V has fixed routes | Waiting time;  Walking link;  Transfer for X(p)  Timetable |
| Bus operation | Y(v) is not stable/reliable, affect Y(p) | Waiting time;  Walking link,  Transfers for X(p)  Bus Service network design for X(v)  Timetable |
| Taxi | Real time matching P and V | Affected by traffic congestion, capacity;  Solve assignment P, V dynamically;  Pick-up and delivery |
| Driving only | V carries P | Traffic assignment problem with UE objectives;  Focus on Y(v) and routing X(v),  Vehicle capacity and road capacity  Computational graph |
| Bike | P carries V | Routing is decided by P |
| Freight commodities | P is package with different load;  V is truck | Pick-up and delivery  Vehicle capacity |





As a simplified or “student” version of DTALite on a fully discretized space-time network, DTALite-S has the following distinctions listed in Table 1.

**Table 1 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 method | Discrete-time in physical network with fixed/regular time interval (6 seconds) | Discrete-time in space-time network with flexible simulation time interval from 0.1 seconds to 1 minutes |

# Conclusions

This research focus on build a modeling framework to bridge simulation and optimization to solve complex transportation problem, along with multimodal transit, VRP, urban rail transit, bus operations. Using mathematic formula to describe the simulation problem from the physical transportation network to a space-time event network. Specifically, the framework addresses several fundamental research issues in large-scale dynamic traffic assignment, mesoscopic traffic simulation and vehicle route optimization. Furthermore, the open-source framework DTALite-S can be embedded in a Lagrangian relaxation and ADMM to handle many general applications.

In the future, a more perfect and microscopic cell-based simulation framework will be further studied, which to simulate more detail behavior, such as lane changing behavior, car flowing behavior and different behavior of drivers. Also, we will focus on the intersection and signal simulation and optimization. The challenge is to control, coordinate, design protocols and analyze operations/ performance over such network.

# References

[1] Zhou X, Taylor J. DTALite: A queue-based mesoscopic traffic simulator for fast model evaluation and calibration[J]. Cogent Engineering, 2014, 1(1):961345.

[2] Newell, G. F. (1993). A simplified theory on kinematic waves in highway traffic, part I: general theory. Transportation Research Part B 27(4), 281-287.

[3] Kirill G, Aleksandr Z, Andrey K. A framework for a multi-agent traffic simulation using combined behavioural models[J]. Procedia Computer Science 136(2018) 443-452.

[4] Yu Y, El Kamel A, Gong G, et al. Multi-agent based modeling and simulation of microscopic traffic in virtual reality system[J]. Simulation Modelling Practice & Theory, 2014, 45(6):62-79.

[5] Haman I T , Kamla V C , Stéphane Galland, et al. Towards an Multilevel Agent-based Model for Traffic Simulation[J]. Procedia Computer Science, 2017, 109:887-892.

[6] Doniec A, Piechowiak S. A behavioral multi-agent model for road traffic simulation[J]. Engineering Applications of Artificial Intelligence, 2008, 21(8):1443-1454.

[7] Tong L, Zhou X, Miller H J, et al. Transportation network design for maximizing space-time accessibility[J]. Transportation Research Part B, 2015, 81:555-576.

[8] Nedic A, Ozdaglar A. Distributed Subgradient Methods for Multi-Agent Optimization[J]. IEEE Transactions on Automatic Control, 2009, 54(1):48-61.

[9] Boyd S, Parikh N, Chu E. Distributed Optimization and Statistical Learning Via the Alternating Direction Method of Multipliers[J]. Foundations & Trends in Machine Learning, 2010, 3(1):1-122.

[10] Nishi T, Ando M, Konishi M. Distributed route planning for multiple mobile robots using an augmented Lagrangian decomposition and coordination technique[J]. IEEE Transactions on Robotics, 2005, 21(6):1191-1200.

Multiagent Modelling and Simulation of a Physical Internet Enabled Rail-Road Intermodal Transport System

Open-source VRPLite Package for Vehicle Routing with Pickup and Delivery: A Path Finding Engine for Scheduled Transportation Systems