



UNIVERSITY OF  
MARYLAND

---

## National Transportation Center

Project ID: NTC2015-MU-R-02

# DEVELOPING A MULTI-RESOLUTION TRAFFIC SIMULATION PLATFORM FOR INTEGRATED ACTIVE TRAFFIC OPERATIONS EVALUATION IN METROPOLITAN AREAS

## Final Report

by

Pitu Mirchandani  
pitu@asu.edu

School of Computing, Informatics, and Decision Systems Engineering  
Arizona State University

Xuesong Zhou  
[xzhou74@asu.edu](mailto:xzhou74@asu.edu), 480-965-5827

School of Sustainable Engineering and the Built Environment  
Arizona State University

Jiangao Liu  
School of Sustainable Engineering and the Built Environment  
Arizona State University

for

National Transportation Center at Maryland (NTC@Maryland)  
1124 Glenn Martin Hall  
University of Maryland  
College Park, MD 20742

**February, 2018**



## **ACKNOWLEDGEMENTS**

This project was funded by the National Transportation Center @ Maryland (NTC@Maryland), one of the five National Centers that were selected in this nationwide competition, by the Office of the Assistant Secretary for Research and Technology (OST-R), U.S. Department of Transportation (US DOT)

## **DISCLAIMER**

The contents of this report reflect the views of the authors, who are solely responsible for the facts and the accuracy of the material and information presented herein. This document is disseminated under the sponsorship of the U.S. Department of Transportation University Transportation Centers Program in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof. The contents do not necessarily reflect the official views of the U.S. Government. This report does not constitute a standard, specification, or regulation.



# TABLE OF CONTENTS

|  |           |
|--|-----------|
| <b>EXECUTIVE SUMMARY .....</b>   | <b>1</b>  |
| <b>1.0 INTRODUCTION.....</b>   | <b>3</b>  |
| <b>2.0 SIMULATION PLATFORM FOR THE INTEGRATION OF AGENT-BASED<br/>ACTIVITY-BASED MODEL AND DYNAMIC TRAFFIC ASSIGNMENT.....</b>                 | <b>7</b>  |
| 2.1 THE PROCEDURE OF DATA COMMUNICATION.....   | 7         |
| 2.2 MODELING AND IMPLEMENTATION FRAMEWORK.....   | 8         |
| 2.2.1 Relation between Data Hub and Data Bus Modules.....  | 8         |
| 2.2.2 Needs for Synchronizing Simulation Clock .....   | 9         |
| 2.2.3 Synchronizing Simulation Clock by Using Scheduled Files as Software File Token   | 9         |
| 2.2.4 Data Flow Chart for Each Process in the Integrated Framework .....   | 10        |
| 2.3 DATA FILE DESCRIPTIONS.....  | 11        |
| <b>3.0 MATHEMATICAL MODELS FOR THE INTEGRATION OF HOUSEHOLD-<br/>LEVEL ABM AND DTA.....</b>  | <b>15</b> |
| 3.1 THE PROCEDURE OF DATA COMMUNICATION.....   | 16        |
| 3.1.1 Network Construction and Conceptual Illustration of Case A .....   | 17        |
| 3.1.2 Network Construction and Conceptual Illustration of Case A .....   | 19        |
| 3.1.3 Conceptual Illustration of Case C .....  | 23        |
| 3.2 MATHEMATICAL PROGRAMMING MODELS .....  | 25        |
| 3.2.1 Space-time Network-based Optimization Model for Case A .....   | 25        |
| 3.2.2 Space-time-state Network-based Optimization Model for Case B.....  | 27        |
| 3.2.3 Link Capacity Constraints of Case C.....   | 30        |
| <b>4.0 NUMERICAL EXPERIMENTS .....</b>   | <b>31</b> |
| 4.1. Small-scale experiment for Case A.....  | 31        |
| 4.2. Small-scale experiment for Case B.....  | 32        |
| 4.3. Medium-scale experiment within a Lagrangian relaxation framework using<br>cumulative activity-performing state .....                      | 35        |
| 4.4. Large-scale experiment within a simulation-based framework with simplified<br>activity representation and road capacity constraints ..... | 37        |
| <b>5.0 CONCLUSIONS .....</b>   | <b>38</b> |

## APPENDICES

APPENDIX A: INSTRUCTIONS AND TIPS FOR USING REPORT TEMPLATE

## LIST OF TABLES

|   |             |
|---|-------------|
| Table 1: Example Data flow of Individual DTA or ABM Module within Agent+ Data Bus. ....                       | 8           |
| Table 2: Notation list for proposed mathematical models. ....   | 15          |
| Table 3: Comparison of model building between Mahmoudi and Zhou (2016) and our case B..                       | 21          |
| Table 4: The maximum number of possible states corresponds with the number of activities in one vehicle. .... | 21          |
| Table 5: Result analysis of different cases. ....   | 2 错误!未定义书签。 |
| Table 6: Comparison between Case IV (Recker, 1995) and our Case A. ....                                       | 26          |
| Table 7: The specific time window for each event. ....  | 38          |
| Table 8: The optimal solution for each household member. ....   | 3 错误!未定义书签。 |
| Table 9: Enumeration of all possible states. ....   | 35          |
| Table 10: The specific time window for each event. ....   | 86          |
| Table 11: The optimal solution for the household. ....  | 37          |
| Table 12: The input data of this experiment. ....   | 38          |
| Table 13: CPU computation time and memory use under different number of activities. ....                      | 39          |

## LIST OF FIGURES

|   |             |
|---|-------------|
| Figure 1: The data flow chat of Agent+ data bus interface for integrating DTA and ABM models. ....        | 8           |
| Figure 2: Illustration of tokens in a distributed computing environment. ....                             | 9           |
| Figure 3: Sample content of Input_simulation_shedule.csv file. ....                                       | 错误!未定义书签。 0 |
| Figure 4: Implemented example of Agent+ framework with connections to data hub. ..                        | 错误!未定义书签。   |
| <b>Figure 1: The data flow chat of Agent+ data bus interface for integrating DTA and ABM models. ....</b> | <b>8</b>    |
| <b>Figure 2: Illustration of tokens in a distributed computing environment. ....</b>                      | <b>9</b>    |
| <b>Figure 3: Sample content of Input_simulation_shedule.csv file. ....</b>                                | <b>10</b>   |
| <b>Figure 4: Implemented example of Agent+ framework with connections to data hub. ....</b>               | <b>11</b>   |
| <b>Figure 5: Sample files from DTALite for OD MOE files and agent lists currently in simulator. ....</b>  | <b>12</b>   |
| <b>Figure 6: Sample attribute list about current agents in DTA simulator. ....</b>                        | <b>12</b>   |
| <b>Figure 1: The data flow chat of Agent+ data bus interface for integrating DTA and ABM models. ....</b> | <b>8</b>    |
| <b>Figure 2: Illustration of tokens in a distributed computing environment. ....</b>                      | <b>9</b>    |
| <b>Figure 3: Sample content of Input_simulation_shedule.csv file. ....</b>                                | <b>10</b>   |
| <b>Figure 4: Implemented example of Agent+ framework with connections to data hub. ....</b>               | <b>11</b>   |
| <b>Figure 5: Sample files from DTALite for OD MOE files and agent lists currently in simulator. ....</b>  | <b>12</b>   |
| <b>Figure 6: Sample attribute list about current agents in DTA simulator. ....</b>                        | <b>12</b>   |
| <b>Figure 1: The data flow chat of Agent+ data bus interface for integrating DTA and ABM models. ....</b> | <b>8</b>    |
| <b>Figure 2: Illustration of tokens in a distributed computing environment. ....</b>                      | <b>9</b>    |

|  |    |
|--|----|
| <b>Figure 3: Sample content of Input_simulation_shedule.csv file.</b>                                | 10 |
| <b>Figure 4: Implemented example of Agent+ framework with connections to data hub.</b>               | 11 |
| <b>Figure 5: Sample files from DTALite for OD MOE files and agent lists currently in simulator.</b>  | 12 |
| <b>Figure 6: Sample attribute list about current agents in DTA simulator.</b>                        | 12 |
| 7  |    |
| <b>Figure 1: The data flow chat of Agent+ data bus interface for integrating DTA and ABM models.</b> | 8  |
| <b>Figure 2: Illustration of tokens in a distributed computing environment.</b>                      | 9  |
| <b>Figure 3: Sample content of Input_simulation_shedule.csv file.</b>                                | 10 |
| <b>Figure 4: Implemented example of Agent+ framework with connections to data hub.</b>               | 11 |
| <b>Figure 5: Sample files from DTALite for OD MOE files and agent lists currently in simulator.</b>  | 12 |
| <b>Figure 6: Sample attribute list about current agents in DTA simulator.</b>                        | 12 |
|  |    |
| <b>Figure 1: The data flow chat of Agent+ data bus interface for integrating DTA and ABM models.</b> | 8  |
| <b>Figure 2: Illustration of tokens in a distributed computing environment.</b>                      | 9  |
| <b>Figure 3: Sample content of Input_simulation_shedule.csv file.</b>                                | 10 |
| <b>Figure 4: Implemented example of Agent+ framework with connections to data hub.</b>               | 11 |
| <b>Figure 5: Sample files from DTALite for OD MOE files and agent lists currently in simulator.</b>  | 12 |
| <b>Figure 6: Sample attribute list about current agents in DTA simulator.</b>                        | 12 |
|  |    |
| <b>Figure 1: The data flow chat of Agent+ data bus interface for integrating DTA and ABM models.</b> | 8  |
| <b>Figure 2: Illustration of tokens in a distributed computing environment.</b>                      | 9  |
| <b>Figure 3: Sample content of Input_simulation_shedule.csv file.</b>                                | 10 |
| <b>Figure 4: Implemented example of Agent+ framework with connections to data hub.</b>               | 11 |
| <b>Figure 5: Sample files from DTALite for OD MOE files and agent lists currently in simulator.</b>  | 12 |
| <b>Figure 6: Sample attribute list about current agents in DTA simulator.</b>                        | 12 |
| 20   |    |
| <b>Figure 1: The data flow chat of Agent+ data bus interface for integrating DTA and ABM models.</b> | 8  |
| <b>Figure 2: Illustration of tokens in a distributed computing environment.</b>                      | 9  |
| <b>Figure 3: Sample content of Input_simulation_shedule.csv file.</b>                                | 10 |
| <b>Figure 4: Implemented example of Agent+ framework with connections to data hub.</b>               | 11 |
| <b>Figure 5: Sample files from DTALite for OD MOE files and agent lists currently in simulator.</b>  | 12 |
| <b>Figure 6: Sample attribute list about current agents in DTA simulator.</b>                        | 12 |
|  |    |
| <b>Figure 1: The data flow chat of Agent+ data bus interface for integrating DTA and ABM models.</b> | 8  |
| <b>Figure 2: Illustration of tokens in a distributed computing environment.</b>                      | 9  |
| <b>Figure 3: Sample content of Input_simulation_shedule.csv file.</b>                                | 10 |

|   |    |
|---|----|
| <b>Figure 4: Implemented example of Agent+ framework with connections to data hub.</b>              | 11 |
| <b>Figure 5: Sample files from DTALite for OD MOE files and agent lists currently in simulator.</b> | 12 |
| <b>Figure 6: Sample attribute list about current agents in DTA simulator.</b>                       | 12 |

|  |    |
|--|----|
| <b>Figure 1: The data flow chat of Agent+ data bus interface for integrating DTA and ABM models.</b> | 8  |
| <b>Figure 2: Illustration of tokens in a distributed computing environment.</b>                      | 9  |
| <b>Figure 3: Sample content of Input_simulation_shedule.csv file.</b>                                | 10 |
| <b>Figure 4: Implemented example of Agent+ framework with connections to data hub.</b>               | 11 |
| <b>Figure 5: Sample files from DTALite for OD MOE files and agent lists currently in simulator.</b>  | 12 |
| <b>Figure 6: Sample attribute list about current agents in DTA simulator.</b>                        | 12 |

23

|  |    |
|--|----|
| <b>Figure 1: The data flow chat of Agent+ data bus interface for integrating DTA and ABM models.</b> | 8  |
| <b>Figure 2: Illustration of tokens in a distributed computing environment.</b>                      | 9  |
| <b>Figure 3: Sample content of Input_simulation_shedule.csv file.</b>                                | 10 |
| <b>Figure 4: Implemented example of Agent+ framework with connections to data hub.</b>               | 11 |
| <b>Figure 5: Sample files from DTALite for OD MOE files and agent lists currently in simulator.</b>  | 12 |
| <b>Figure 6: Sample attribute list about current agents in DTA simulator.</b>                        | 12 |

|  |    |
|--|----|
| <b>Figure 1: The data flow chat of Agent+ data bus interface for integrating DTA and ABM models.</b> | 8  |
| <b>Figure 2: Illustration of tokens in a distributed computing environment.</b>                      | 9  |
| <b>Figure 3: Sample content of Input_simulation_shedule.csv file.</b>                                | 10 |
| <b>Figure 4: Implemented example of Agent+ framework with connections to data hub.</b>               | 11 |
| <b>Figure 5: Sample files from DTALite for OD MOE files and agent lists currently in simulator.</b>  | 12 |
| <b>Figure 6: Sample attribute list about current agents in DTA simulator.</b>                        | 12 |

|  |    |
|--|----|
| <b>Figure 1: The data flow chat of Agent+ data bus interface for integrating DTA and ABM models.</b> | 8  |
| <b>Figure 2: Illustration of tokens in a distributed computing environment.</b>                      | 9  |
| <b>Figure 3: Sample content of Input_simulation_shedule.csv file.</b>                                | 10 |
| <b>Figure 4: Implemented example of Agent+ framework with connections to data hub.</b>               | 11 |
| <b>Figure 5: Sample files from DTALite for OD MOE files and agent lists currently in simulator.</b>  | 12 |
| <b>Figure 6: Sample attribute list about current agents in DTA simulator.</b>                        | 12 |

|  |    |
|--|----|
| <b>Figure 1: The data flow chat of Agent+ data bus interface for integrating DTA and ABM models.</b> | 8  |
| <b>Figure 2: Illustration of tokens in a distributed computing environment.</b>                      | 9  |
| <b>Figure 3: Sample content of Input_simulation_shedule.csv file.</b>                                | 10 |
| <b>Figure 4: Implemented example of Agent+ framework with connections to data hub.</b>               | 11 |



|  |    |
|--|----|
| <b>Figure 5: Sample files from DTALite for OD MOE files and agent lists currently in simulator.</b> .....  | 12 |
| <b>Figure 6: Sample attribute list about current agents in DTA simulator.</b> .....                        | 12 |
| <b>Figure 1: The data flow chat of Agent+ data bus interface for integrating DTA and ABM models.</b> ..... | 8  |
| <b>Figure 2: Illustration of tokens in a distributed computing environment.</b> .....                      | 9  |
| <b>Figure 3: Sample content of Input_simulation_shedule.csv file.</b> .....                                | 10 |
| <b>Figure 4: Implemented example of Agent+ framework with connections to data hub.</b> .....               | 11 |
| <b>Figure 5: Sample files from DTALite for OD MOE files and agent lists currently in simulator.</b> .....  | 12 |
| <b>Figure 6: Sample attribute list about current agents in DTA simulator.</b> .....                        | 12 |
| <b>Figure 1: The data flow chat of Agent+ data bus interface for integrating DTA and ABM models.</b> ..... | 8  |
| <b>Figure 2: Illustration of tokens in a distributed computing environment.</b> .....                      | 9  |
| <b>Figure 3: Sample content of Input_simulation_shedule.csv file.</b> .....                                | 10 |
| <b>Figure 4: Implemented example of Agent+ framework with connections to data hub.</b> .....               | 11 |
| <b>Figure 5: Sample files from DTALite for OD MOE files and agent lists currently in simulator.</b> .....  | 12 |
| <b>Figure 6: Sample attribute list about current agents in DTA simulator.</b> .....                        | 12 |
| <b>Figure 1: The data flow chat of Agent+ data bus interface for integrating DTA and ABM models.</b> ..... | 8  |
| <b>Figure 2: Illustration of tokens in a distributed computing environment.</b> .....                      | 9  |
| <b>Figure 3: Sample content of Input_simulation_shedule.csv file.</b> .....                                | 10 |
| <b>Figure 4: Implemented example of Agent+ framework with connections to data hub.</b> .....               | 11 |
| <b>Figure 5: Sample files from DTALite for OD MOE files and agent lists currently in simulator.</b> .....  | 12 |
| <b>Figure 6: Sample attribute list about current agents in DTA simulator.</b> .....                        | 12 |
| <b>Figure 1: The data flow chat of Agent+ data bus interface for integrating DTA and ABM models.</b> ..... | 8  |
| <b>Figure 2: Illustration of tokens in a distributed computing environment.</b> .....                      | 9  |
| <b>Figure 3: Sample content of Input_simulation_shedule.csv file.</b> .....                                | 10 |
| <b>Figure 4: Implemented example of Agent+ framework with connections to data hub.</b> .....               | 11 |
| <b>Figure 5: Sample files from DTALite for OD MOE files and agent lists currently in simulator.</b> .....  | 12 |
| <b>Figure 6: Sample attribute list about current agents in DTA simulator.</b> .....                        | 12 |



## EXECUTIVE SUMMARY

To be able to understand and model the implications of the advanced technologies and their impact on traveler behavior, it is vital to have integrated model systems that fully capture the interactions between supply and demand dimensions of travel. This research proposes a multi-resolution traffic simulation platform which integrate (1) an activity-based travel demand model, referred to as AgBM which can perform an agent-based microsimulation to predict multi-dimensional travel behaviors and (2) a simulation-based dynamic traffic assignment tool, referred to as DTALite, to conduct large-scale traffic operations that have active traffic management strategies as ingredients. As a result, the traffic congestion and feedback loops associated with complex trip interactions and human activity-travel decisions are captured.

In addition, the recently emerging trend of self-driving vehicles and information sharing technologies, made available by private technology vendors, starts creating a revolutionary paradigm shift in the coming years for traveler mobility applications. By considering a deterministic traveler decision making framework at the household level in congested transportation networks, this paper aims to address the challenges of how to optimally schedule individuals' daily travel patterns under the complex activity constraints and interactions. We reformulate two special cases of household activity pattern problem (HAPP) through a high-dimensional network construct, and offer a systematic comparison with the classical mathematical programming models proposed by Recker (1995). Furthermore, we consider the tight road capacity constraint as another special case of HAPP to model complex interactions between multiple household activity scheduling decisions, and this attempt offers another household-based framework for linking activity-based model (ABM) and dynamic traffic assignment (DTA) tools. Through embedding temporal and spatial relations among household members, vehicles and mandatory/optional activities in an integrated space-time-state network, we develop two 0-1 integer linear programming models that can seamlessly incorporate constraints for a number of key decisions related to vehicle selection, activity performing and ridesharing patterns under congested networks. The well-structured network models can be directly solved by standard optimization solvers, and further converted to a set of time-dependent state-dependent least cost path-finding problems through Lagrangian relaxation, which permit the use of computationally efficient algorithms on large-scale high-fidelity transportation networks.



## 1.0 INTRODUCTION

Metropolitan areas count for a major portion of economic activities in US and it is critical to maintain and improve the health of transportation systems in metropolitan areas so as to increase the economic competitiveness. Maintaining and operating metropolitan transportation systems is challenging because of the lack of effective models and analysis tools to: (1) estimate and predict travel demand; and (2) represent/analyze the complex interactions between traffic management and driver behaviors. In the past, these challenges are usually addressed separately. Therefore, it is necessary to develop a new integrated tool to synergistically address the issues of travel behaviors and travel decision-making, traffic assignment and traffic operations. This report is for the development of a simulation platform, where multi-resolution simulation will be enhanced with smart travelers, information sharing, and household-level daily scheduled activities.

Great strides have been made in the past couple of decades in advancing travel demand modeling from the traditional 4-step travel demand models where the demand and supply sides were considered static to present day state-of-the art integrated travel model systems. On the travel demand front, the profession has progressed from traditional trip-based methods to activity-based models (ABMs), which date back to the pioneering work of Kitamura (1998) (see Rasouli et al. (2014) for a detailed synthesis on ABMs). ABMs view travel as derived demand, arising from the necessity of individuals to participate in various activities. This facilitates representing travel in a behaviorally realistic way in ABMs. On the other hand, network supply/simulation has progressed from static traffic assignment to dynamic traffic assignment (DTA) models that employ microscopic simulation and are capable of evaluating various traffic management strategies on the fly.

To consider the traffic congestion and feedback loops associated with complex trip interactions, there are a wide range of studies aiming to combine ABM and DTA to better capture the interplay between human activity-travel decisions and underlying congested networks with tight road capacity constraints. For example, Lin et al. (2008) proposed a conceptual framework and explored the model integration of activity-based model (CEMDAP) and dynamic traffic assignment model (VISTA). Pendyala et al. (2012) further integrated activity-travel demand models (OpenAMOS), DTA tools with the long-term land use modeling layer (UrbanSim). To further study the impacts of dynamic traffic management strategies and real-time traveler information provision, Pendyala et al. (2017) proposed a tightly integrated modeling framework for representing activity-travel demand and traffic dynamics in an on-line environment.

In this report, a new simulation platform for integrating ABM and DTA models is proposed for integrated active traffic operations evaluation in metropolitan areas. It allows minute-by-minute data conversion and communication utilities between different ABM and DTA packages, and supports comprehensive and streamlined analysis of the complex transportation system environment to evaluate a wide range of ABM+DTA scenarios, such as, dynamic pricing, dynamic flow control, and dynamic information provision.

Further, the past decade has witnessed unprecedented advancements in the auto industry, specifically in the domain of autonomous vehicle technologies. Several auto companies have forged new paths and introduced vehicles of the future that need minimal human intervention for their operation (Tesla Motors Team (2016); Sherman (2016)). Ridesourcing, operated by Transportation Network Companies (TNCs) such as Uber and Lyft, is another game changing technology introduced in the recent times. TNCs aim at providing reliable and inexpensive personalized travel options that combine the best of personalized transport (for example, door-to-door travel), as well as transit services (where the users pay per trip, and also do not have to drive the vehicle themselves). Recent reports show that 12% of registered voters across the United States used ridesourcing services at least once in the past month (Morning Consult (2015)).

The rapidly growing popularity of TNCs coupled with autonomous vehicle technologies, could potentially redefine the way in which individuals schedule their activities and also the way in which travel demand is managed by network operators. On the traveler's front, the freedom from driving could mean flexible activity schedules and increased productivity while travelling. On the other hand, network operators could handle demand by incentivizing/dis-incentivizing travel during a certain portion of the day (similar to surge pricing by Uber), or using a specific route. There is growing interest in the field to study incentive-based demand management strategies (for example, see Hu et al. (2014)). It is therefore of critical importance to understand and accurately depict these transformative technologies and their implications on activity-travel patterns in the travel demand model systems. While the integrated models developed so far cater to modeling the current array of travel options (modes, demand management strategies, etc.), most of the integrated model systems are not capable of handling the emerging transportation technologies (ride-sharing services, autonomous vehicle technologies) that are going to become a reality in the near future.

Specifically, ABMs still operate based on zonal level information (such as skims, by time-of-day) provided by DTA models. The ABMs are oblivious to network logistics such as availability of ridesourcing options and incentives/disincentives customized for specific trips/travelers. On the other hand, vehicle routing problems (VRP), used to depict ride-sharing services in DTA models, view travel as disjoint trips that are independent of each other. The solutions to VRP are typically optimization-based and lack a sound behavioral foundation (Cordeau et al. (2001)). Solutions to VRP in the standard DTA models are often aimed at serving the maximum number of trip requests without taking into consideration the precedence constraints (or linkages) between the trips. Consider an individual's schedule comprising of three trips: a) pick-up his child, b) accompany the child to the playground and c) take the child home. A VRP algorithm could produce a solution where trip requests for activities 'b' and 'c' are served, but in reality, activities 'b', and 'c' have a precedence constraint of engaging in activity 'a'. This vital behavioral constraint is ignored in the VRP optimization techniques operated in DTA models.

More importantly, at the household level, how to recognize complex resource constraints, multi-agent interactions, and consistency through trip chains of different individuals is an important concern for accurate activity-based modelling and analysis. Different modeling paradigms have been developed, including deterministic optimization-based models by Recker (1995), and probabilistic micro-simulation-based utility maximization models by Bhat et al. (2004), Pendyala et al. (2005), Pribyl and Goulias (2005), Miller and Roorda (2003), and Arentze and Timmermans (2004). The emerging mobile apps with multi-modal traveler information and personal activity schedules enable travelers to intelligently schedule their activities and share their trip requests. In

addition, the forthcoming autonomous vehicle system would allow and encourage a fully optimized planning process for mapping household activities and travel requests (to be met by personal or shared vehicles). Therefore, this study also focus on the household activity pattern problem (HAPP) that is first systematically formulated by Recker (1995), which aims to find the optimal path of household members for completing their prescribed activities based on the available number of vehicles, scheduled activity participation, and ride-sharing options within a long period as the unit of analysis.

Typically, based on a conventional mixed integer linear programming model for the pickup and delivery problem with time windows (PDPTW), many typical cases in HAPP, e.g., five cases in a classical paper by Recker (1995), Recker (2001), Recker et al. (2001), and Gan and Recker (2008), require a very large number of linear and integer constraints to capture the complex rules in real-world household-level activity scheduling progress. Recently, several algorithms had been proposed to address more realistic side constraints and large-sized examples, to name a few, Chow and Recker (2012) and Kang and Recker (2013). In addition, Liao et al. (2013a, 2013b) presented a new set of super-network models for various person-level activity scheduling problems, where the multi-dimensional network construct contains travel links, state transition links and activity transaction links. To formulate HAPP as a mathematically rigorous model, how to fully consider complex coupling constraints among three layers, namely household members, vehicles and mandatory/optional activities, is extremely challenging, especially for large-scale multi-modal transportation network with flexible ride-sharing and household member activity-coordination options.

Based on mathematical programs of HAPP, Kang et al. (2013) studied the network design problem considering the interaction between the household-level activity pattern and infrastructure changes. Chow and Djavadian (2015) proposed a new market equilibrium model to capture the interaction of traveler activity schedules in a capacitated system with a macroscopic flow restriction on a link or node facility. Abdul Aziz and Ukkusuri (2013) examined capacitated vehicle routing problems with the time-dependent congestion costs, which are determined by a network-wide cell transmission model. In a recent study by Fu et al. (2016), the intra-household interactions are considered through Markov decision processes and the road congestion effect is reflected by the static travel time function.

In this report, we aims to cast HAPP problems as number of time-dependent and state-dependent path searching problems, which have a class of computationally efficient algorithms available in discretized space-time network and high-dimensional space-time-state networks. To capture the impacts of traffic congestion on activity generation and scheduling, this paper also reformulates two special cases of HAPPs as system-optimal multi-household activity scheduling subject to the tight road capacity constraints. The key is how to prebuild a set of embedded finite state machines (FSM) in a network to precisely represent and translate side constraints from the traditional models, which could eliminate activity time window and vehicle selection constraints in the resulting optimization model. Specifically, we consider Case A as Multi-vehicle and Multi-person vehicle routing problem with mandatory and discretionary activities. Further, with the given ride-sharing options for each household, we propose one more dimension to represent the activity performing status in each vehicle and model our Case B as Multi-vehicle and Multi-person ridesharing problem with mandatory and discretionary activities. These two problems can be formulated as 0-1 integer linear programming models, with the space-time-state network being indexed through

vehicle's location, vehicle's timestamp and cumulative activity completion state. Then the road capacity constraint can be directly added to model the network congestion and resulting activity scheduling change. Through dualizing the capacity constraints to the objective function by Lagrangian relaxation, our proposed model can be further solved through time-dependent state-dependent least cost path-finding algorithms, which permits the use of fast computational algorithms on large-scale high-fidelity transportation networks.



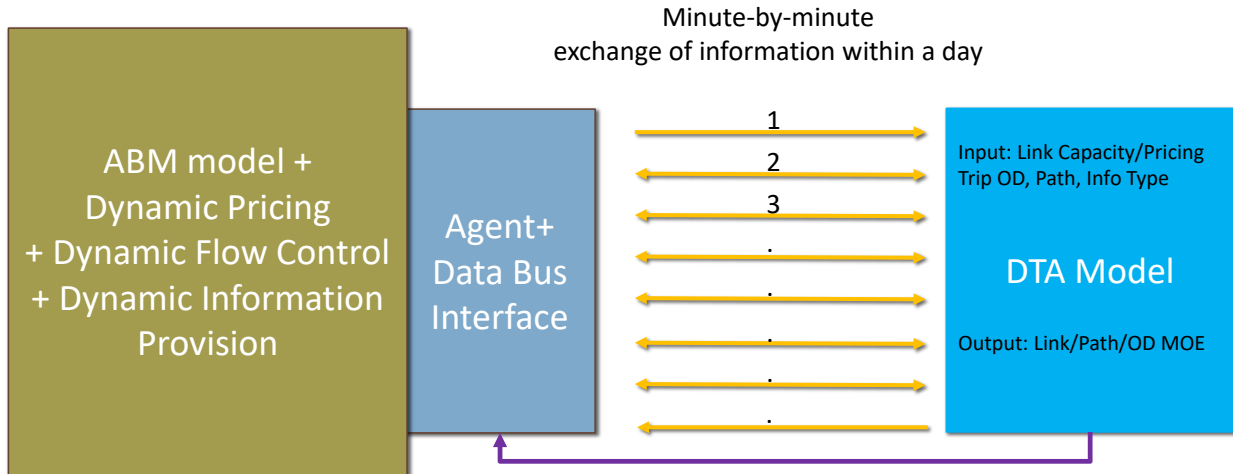
## **2.0 SIMULATION PLATFORM FOR THE INTEGRATION OF AGENT-BASED ACTIVITY-BASED MODEL AND DYNAMIC TRAFFIC ASSIGNMENT**

The proposed multi-resolution traffic simulation platform mainly includes an agent-based travel demand model, referred to as AgBM which can performance an agent-based microsimulation to predicts multi-dimensional travel behaviors, and a simulation-based dynamic traffic assignment tool, referred to as DTALite to conduct large-scale traffic operations that have active traffic management strategies as ingredients.

Specifically, AgBM provides DTALite only time-dependent origin-destination (OD) pair telling DTALite the departure times and destinations of agents. Then the DTALite engine will be launched to generate the core simulation output files for AgBM, including link-level measures of effectiveness (MOE), agent trajectory, etc. These simulation output files provide necessary information for AgBM to predict reasonable travel (agent) behavior changes in response to real-time traffic conditions and active traffic management strategies, such as new departure time and re-routing decisions by each traveler. AgBM will predict travelers' pre-trip and dynamic route decisions based on observed current conditions. These AgBM outputs will in turn be fed into DTALite again as input files to update the DTALite simulation configuration in a real-time manner. AgBM defines convergence criteria as a dynamic behavioral equilibrium, i.e. all simulated agents stop making any further behavior changes. This equilibria condition is also fed to DTALite to produce the final outputs for the integrated model. As a result, after the AgBM is integrated with DTALite, the travelers in DTALite will have more "intelligence" in their learning and decision making process. For instance, an AgBM-enabled traveler in DTALite will be able to determine the departure time according to multiple factors. Also the traveler can change path choice en-route not only based on the minimum travel cost but also many other factors according to the latest traffic information. These new features reflect the latest reality and will definitely provide the decision makers with more convincing proofs to help analyze active traffic management strategies.

### **2.1 THE PROCEDURE OF DATA COMMUNICATION**

Many of the traffic analysis tools (e.g. VISSIM, TransModeler, Aimsun, and Paramics) have proprietary components that could create impediments for exchanging data across independent software packages that require the development and application of individual utility functions. While the actual solution will be derived during the project execution, our proposed solution to this challenge is to develop an open data format that allows data conversion and communication utilities from their own different formats to the proposed Agent+ data bus. While the framework can be implemented without an agent+ data bus or an open data communication format standard on a data hub, only ad-hoc linkages across specific software packages can be established, which could fail to allow comprehensive and streamlined analysis of the complex transportation system environment to support a wide range of ABM+DTA scenario evaluation.



**Figure 1: The data flow chat of Agent+ data bus interface for integrating DTA and ABM models.**

Fig. 1 above shows the Agent+ Data Bus concept. Our team already implemented an open-source Agent+ data hub that allows a day-by-day and min-by-min integration of ABM and DTA models. New development efforts are needed to implement the data bus for an on-line data communication environment, but can be implemented with minimal risks given our team’s experience in developing the Data hub.

**Table 1: Example Data flow of Individual DTA or ABM Module within Agent+ Data Bus.**

| Package | Software Packages Connected to Data Bus  | Receive Input from Data Bus  | Provide Output to Data Bus  |
|---------|--|--|---|
| ABM     | CSV file reading and writing utility   | Day-to-day or min-by-min traffic conditions at link/path/OD levels                                     | Min-by-min traffic request for possible departure time, destination, and mode change  |
| DTA     | CSV file reading and writing utility or Data receiving and aggregation package | Traffic network, vehicle origin and destination, existing traveled path, real-time traffic data stream | Information for upstream vehicles to change routes, updated travel time and queue warnings, traffic conditions at link/path/OD levels |

## 2.2 MODELING AND IMPLEMENTATION FRAMEWORK

### 2.2.1 Relation between Data Hub and Data Bus Modules

The Data Hub hosts essential network, demand, signal control data for multi-resolution modeling. The real-world traffic measurements from private data vendors and public agencies can be also archived in the databases.

The Data Bus aims to enable tight interconnections between three critical entities: data, models, and simulators of decision makers. The data bus provides excellent data interchange interoperability which holds a key means of accelerating the integration of DTA+ABM distributed

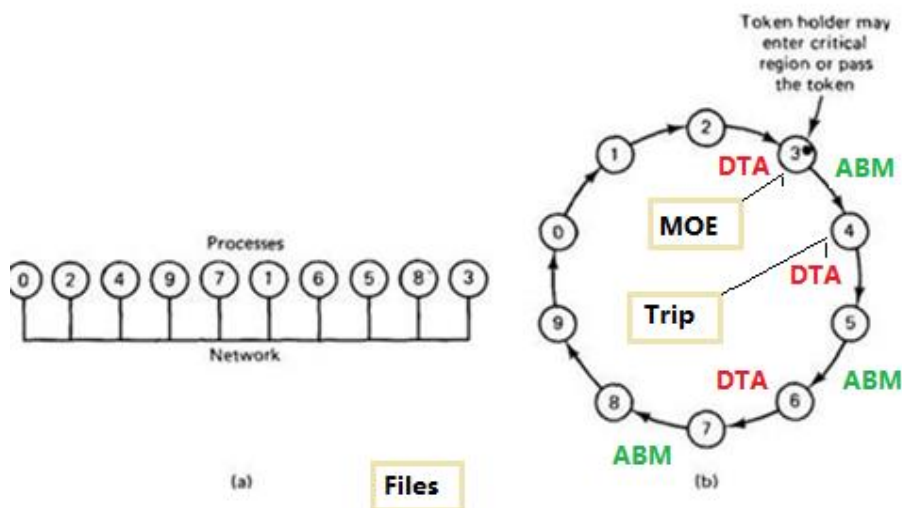
simulation. In the long run, it allows simulation models to be executed through multiple computation instances in a virtual cloud computing center or in a parallel computing environment.

### 2.2.2 Needs for Synchronizing Simulation Clock

Multiple modules (e.g. demand simulator, network simulator) within the multi-resolution analysis framework execute in different processors and communicate with each other asynchronously. Different simulation modules can run with different execution cycles, and each step size is determined by the corresponding simulation needs (e.g. 6 seconds for mesoscopic traffic simulator vs. 1 min for travel demand simulator), computational resource constraints, as well as input/output dependency between different models. The proposed data bus concept further highlights the needs for standardized data format across multi-resolution simulation, a tight coupling with ABM and DTA simulation, and clearly defined rules for performing multi-horizon simulation/prediction for applications such as dynamic flow control, dynamic pricing and dynamic information provision. In the SHRP II Maryland test bed with multiple simulators, it is important to ensure all the computer applications are synchronized with a common simulation clock, while certain high-fidelity simulation systems may be even slower than real-time (e.g., one simulated second is equal to 10 real-world seconds).

### 2.2.3 Synchronizing Simulation Clock by Using Scheduled Files as Software File Token

In general, [source: wiki] a security token (or sometimes a hardware token, authentication token, USB token, cryptographic token, software token, virtual token, or key fob) may be a physical device that an authorized user of computer services is given to ease authentication. In our study, we use files as software tokens.



**Figure 2: Illustration of tokens in a distributed computing environment.**

Each DTA/ABM process must have the file token to move forward, then perform simulation and generate the next token required by the sequential process. The file tokens have fixed names to be better synchronized across processes.

## 2.2.4 Data Flow Chart for Each Process in the Integrated Framework

Read input schedule file to follow the token reading and output frequency and file name specification.

While (for each x second scheduled)

```
{
    If (input file token is ready)
    {
        Perform simulation/analysis
        Generate output file token (for next process)
    }
}
```

Specifically, in the proposed agent+ modeling framework, we will first specify the data updating frequency and file name convention in a file called “Input\_simulation\_schedule.csv”, for output Link/Path/OD MOE files and for input attributes and agent routing attributes. In this example, the input and output data streams are defined with respect to the DTA module.

| attribute     | start_time | end_time | RT_Output_LinkMOE | RT_Output_PathMOE | RT_Output_ODMOE | RT_Output_Current_Agent | RT_Input_LinkAttribute | RT_Input_Update_Agent |
|---------------|------------|----------|-------------------|-------------------|-----------------|-------------------------|------------------------|-----------------------|
| time_interval | 0          | 300      | 60                | 60                | 300             | 6                       | 0                      | 300                   |

**Figure 3: Sample content of Input\_simulation\_schedule.csv file.**

By detecting if the required input files are available every time intervals (say 1 min), the agent+ modeling framework can accordingly use the scheduled input and output files as tokens for synchronizing the simulation clock with the data bus’s master clock.

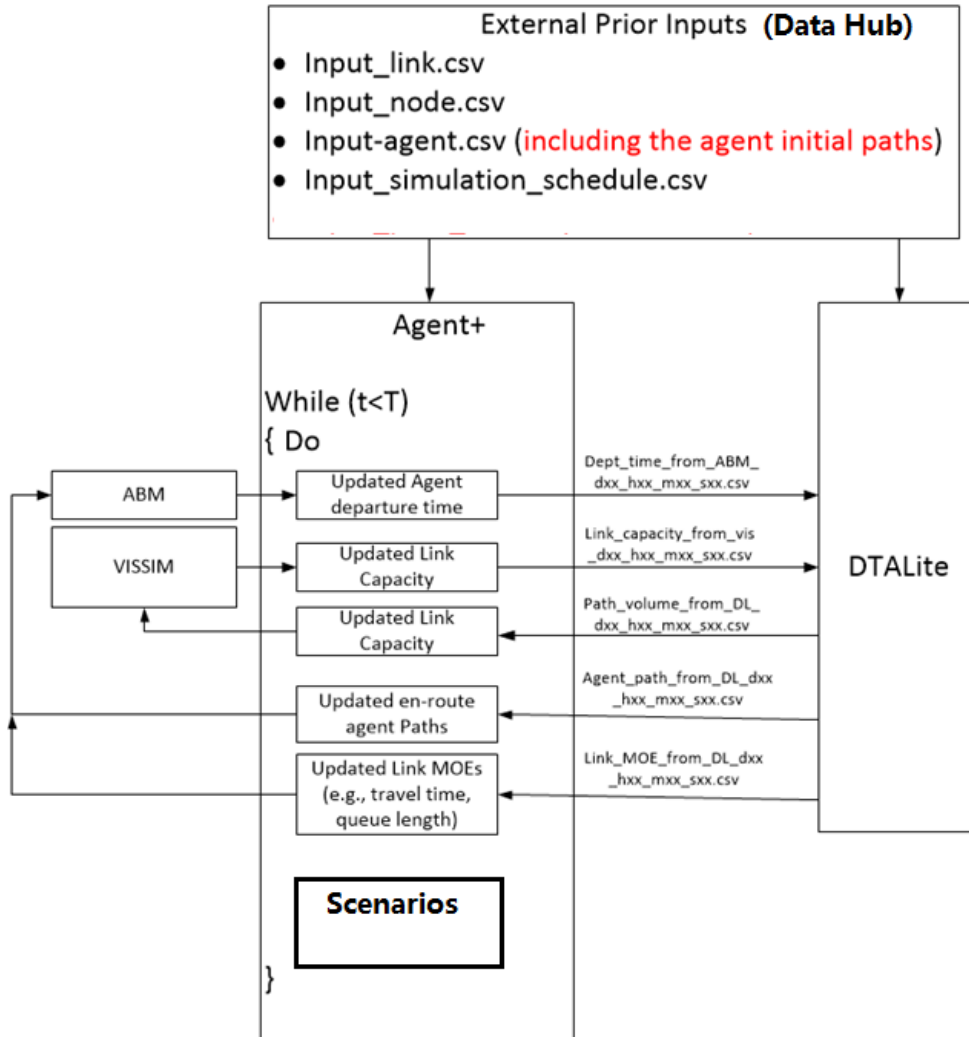
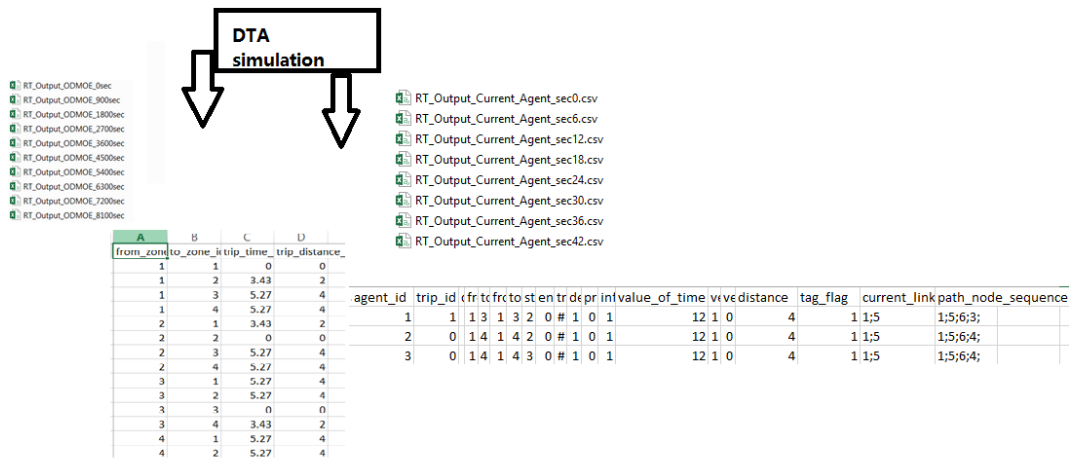


Figure 4: Implemented example of Agent+ framework with connections to data hub.

## 2.3 DATA FILE DESCRIPTIONS

A general example is show in Figure 5 to illustrate those necessary data files in the DTA simulation package.



**Figure 5: Sample files from DTALite for OD MOE files and agent lists currently in simulator.**

Specifically, there are three data blocks:

Data Block 1: MOE output files from DTA

(1) RT\_output\_linkMOE\_secXX.csv

Attributes: from\_node, to\_node, travel time, speed and timestamp at 60-second resolution.

(2) RT\_output\_ODMOE\_secXX.csv

Attributes: zone number, travel time, travel distance and number of agents every 10 or 15 min.

(3) RT\_output\_pathMOE\_secXX.csv

Major attribute is the node sequence for each path at 60-second resolution.

Data block 2: Current agent list output from DTA, at current time or after a whole day

(1) RT\_Output\_Current\_Agent\_secXX.csv

Main 15 attributes for agents currently in the traffic simulator: agent\_id, detour node sequence, pricing type, and value of time, as shown in Figure 6.

| agent_id | trip_id | (fr,tc,frct,rc,tcst,ent,rd,de,pr,inf) | value_of_time | v | ve | distance | tag_flag | current_link | path_node_sequence |
|----------|---------|---------------------------------------|---------------|---|----|----------|----------|--------------|--------------------|
| 1        | 1       | 1 3 1 3 2 0 # 1 0 1                   | 12            | 1 | 0  | 4        | 1 1;5    | 1;5;6;3;     |                    |
| 2        | 0       | 1 4 1 4 2 0 # 1 0 1                   | 12            | 1 | 0  | 4        | 1 1;5    | 1;5;6;4;     |                    |
| 3        | 0       | 1 4 1 4 3 0 # 1 0 1                   | 12            | 1 | 0  | 4        | 1 1;5    | 1;5;6;4;     |                    |

**Figure 6: Sample attribute list about current agents in DTA simulator.**

(2) RT\_Output\_End-of-Day\_Agent\_dayXX.csv

Attributes: Experienced travel times and path for each agent from the previous day:

Data block 3: min-by-min input files for updating DTA status

RT\_Input\_linkAttribute\_secXX.csv

RT\_Input\_Updated\_Agent\_secXX.csv

Input\_routing\_policy.csv

These real-time link attributes file and agent file are updated every 60 seconds, and the above 2 files can be controlled from ABM or updated based on predefined schedules to model.

The updated agent files can allow models to change traveler decisions in terms of departure time, destination, and information type (e.g. from historical, pretrip at origin, to real-time en-route) and route information. Canceling trip option is also allowed by removing agents from the to-be-simulated list or simply delaying the departure time of agents to next day.





### 3.0 MATHEMATICAL MODELS FOR THE INTEGRATION OF HOUSEHOLD-LEVEL ABM AND DTA

The notation used in this section is listed in Table 2.

**Table 2: Example Data flow of Individual DTA or ABM Module within Agent+ Data Bus.**

| Symbols              | Descriptions   |
|----------------------|--|
| $N$                  | Set of nodes in the physical network, including necessary virtual nodes                                  |
| $N_v$                | Set of vehicle nodes for vehicle selection   |
| $L$                  | Set of links in the physical network, including necessary virtual links                                  |
| $P$                  | Set of household members   |
| $P_m$                | Set of household members who have mandatory activities   |
| $P_n$                | Set of household members who chooses one mandatory activity from multiple candidates                     |
| $P_q$                | Set of household members who have discretionary activities   |
| $V$                  | Set of available vehicles  |
| $A$                  | Set of activities  |
| $A_n(p)$             | Set of household member $p$ 's candidate activities for one kind of mandatory activity                   |
| $A_v$                | Set of mandatory activities of vehicle $v$ 's driver   |
| $R$                  | Set of vertices in the space-time/space-time-state network   |
| $E$                  | Set of edges/arcs in the space-time/space-time-state network   |
| $W$                  | Set of cumulative vehicle activity-performing state  |
| $E(p, a_m)$          | Set of edges/arcs of household member $p$ 's mandatory activity $a_m$                                    |
| $E(p, a_n)$          | Set of edges/arcs of household member $p$ 's candidate activity $a_n$ for one kind of mandatory activity |
| $E(p, a_q)$          | Set of edges/arcs of household member $p$ 's discretionary activity $a_q$                                |
| $E(v, a_m)$          | Set of edges/arcs of mandatory activity $a_m$ of vehicle $v$ 's driver                                   |
| $i, j$               | Index of node set $N$  |
| $(i, j)$             | Index of link set $L$  |
| $t, s$               | Index of time intervals in the space-time-state network  |
| $w, w'$              | Index of state in the space-time-state network   |
| $(i, t)$             | Index of vertex in the space-time network  |
| $(i, j, t, s)$       | Index of edges/arcs in the space-time network  |
| $(i, t, w)$          | Index of vertex in the space-time-state network  |
| $(i, j, t, s)$       | Index of edges/arcs in the space-time-state network  |
| $p$                  | Index of household member set $P$  |
| $a$                  | Index of activity set $A$  |
| $t(i, j)$            | Travel time of link $(i, j)$   |
| $c_{i,j,t,s}^p$      | Travel cost of arc $(i, j, t, s)$ of person $p$ in the space-time network                                |
| $c_{i,j,t,s,w,w'}^v$ | Travel cost of arc $(i, j, t, s, w, w')$ of vehicle $v$ in the space-time-state network                  |
| $[a_k, b_k]$         | The time window of event $k$ , such as, activity starting time window, activity ending time window       |
| $TD(p)/TD(v)$        | Earliest departure time of household member $p$ / vehicle $v$  |
| $O(p)/O(v)$          | Origin node of household member $p$ / vehicle $v$  |
| $D(p)/D(v)$          | Destination node of household member $p$ / vehicle $v$   |
| $T$                  | The time horizon in the space-time network/space-time-state network                                      |
| $Cap_{i,j,t,s}$      | Capacity of arc $(i, j, t, s)$   |

|                      |  |
|----------------------|--|
| $Cap_{i,j,t,s,w,w'}$ | Capacity of arc $(i, j, t, s, w, w')$  |
| $x_{i,j,t,s}^p$      | Binary variable, = 1, if household member $p$ visits the traveling/waiting arc $(i, j, t, s)$ in the space-time network; = 0 otherwise     |
| $x_{i,j,t,s,w,w'}^v$ | Binary variable, = 1, if vehicle $v$ visits the traveling/waiting arc $(i, j, t, s, w, w')$ in the space-time-state network; = 0 otherwise |
| $N$                  | Set of nodes in the physical network, including necessary virtual nodes  |
| $N_v$                | Set of vehicle nodes for vehicle selection   |

### 3.1 THE PROCEDURE OF DATA COMMUNICATION

In this paper, our study focuses on three particular cases in HAPP (namely A, B, C) and further extends to one more general case D. The general given input includes the population used for activity generation, a physical transportation network, a set of different types of activities (mandatory, semi-mandatory, optional activities) with specific time windows and utility values, a set of vehicles, as well as the activity/vehicle assignment set to each household member. By adapting the classical assumption/definition from Recker (1995), we present the following problem statements.

(1) Case A is a multi-vehicle and multi-person vehicle routing problem with mandatory and discretionary activities, which is similar to Case IV in the paper by Recker (1995). (i) Members of the household share a set of vehicles; a subset of vehicles may be available for use by any member of the household, and the remainder may be reserved for use by certain members; (ii) A subset of activities can be performed by any member of the household, and the remaining activities must be performed by certain members; (iii) Certain members can have specific mandatory activities or optional activities; (iv) Some members may perform no activities; some vehicles may not be used.

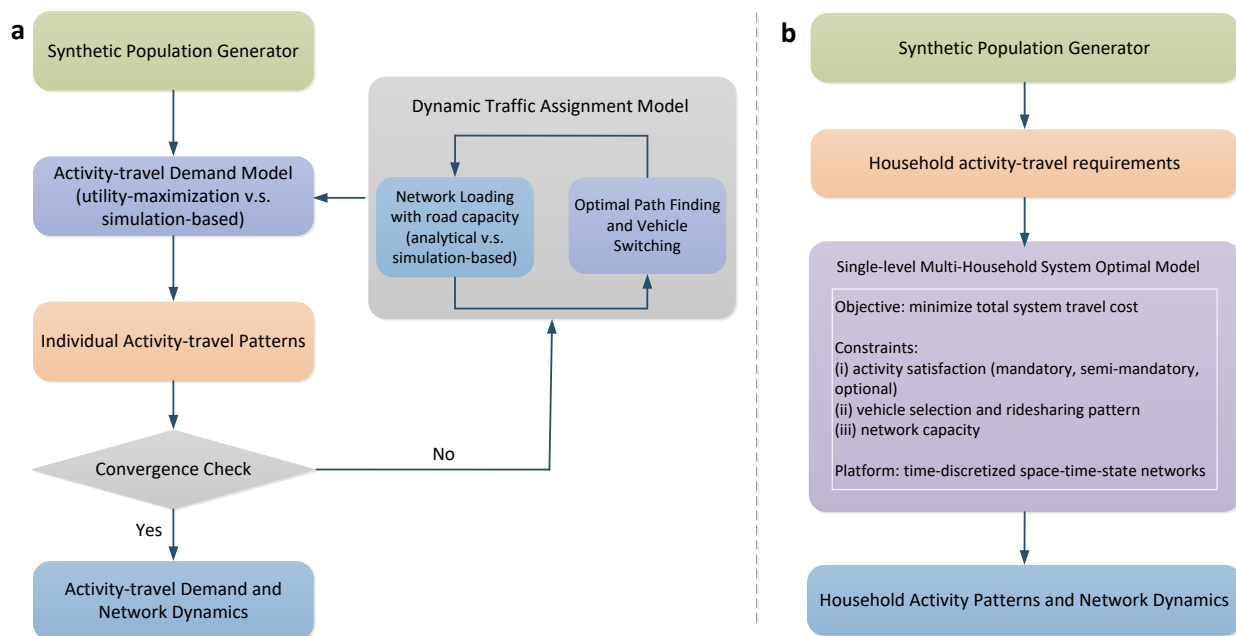
(2) Case B is a multi-vehicle and multi-person ridesharing problem with mandatory and discretionary activities, which can be treated as a special sub-problem of Case V in the paper by Recker (1995). The specific definition is: (i) the ride-sharing pattern that which household members will share one vehicle and which one is the driver has been given; (ii) A subset of activities can be performed by any member of the household, and the remaining activities must be performed by certain members; (iii) Certain members can have specific mandatory activities or optional activities.

(3) Case C is an extension of cases A and B, which considers tight road capacity constraints to capture the underlying congestion in physical transportation networks, so that the influence of time-dependent link travel time on household activity patterns can be observed. As a result, this case is a system optimal multi-household activity scheduling problem under time-varying traffic conditions.

(4) Case D is a dynamic household-level equilibrium problem where each household is inclined to choose the optimal activity pattern, which considers vehicle selection, mode choice and ride-sharing options simultaneously. As studied in a recent paper by Liu and Zhou (2016), when there is no link capacity constraint, each agent (e.g., passenger, vehicle, or household) can choose the best/shortest path without affecting each other. Once the limited resource constraint is strictly

considered, some agents may have to accept a longer path in order to finish their own travel and this kind of decision mechanism could invoke bounded rationality to those agents.

Specially, Fig. 7 specifically compares the data flow of (i) existing integration of ABM and DTA and (ii) our proposed models. The simulation-based integration in Fig. 7 focuses on searching individual activity-travel pattern under dynamic user equilibrium conditions, and the mathematical program oriented modelling framework proposed in this paper aims to optimize the household activity decisions with system optimal goals under household-level activity requirements and network capacity constraints. In our future study, Case D will be further examined to study possible dynamic household-level equilibriums with household activity interactions.



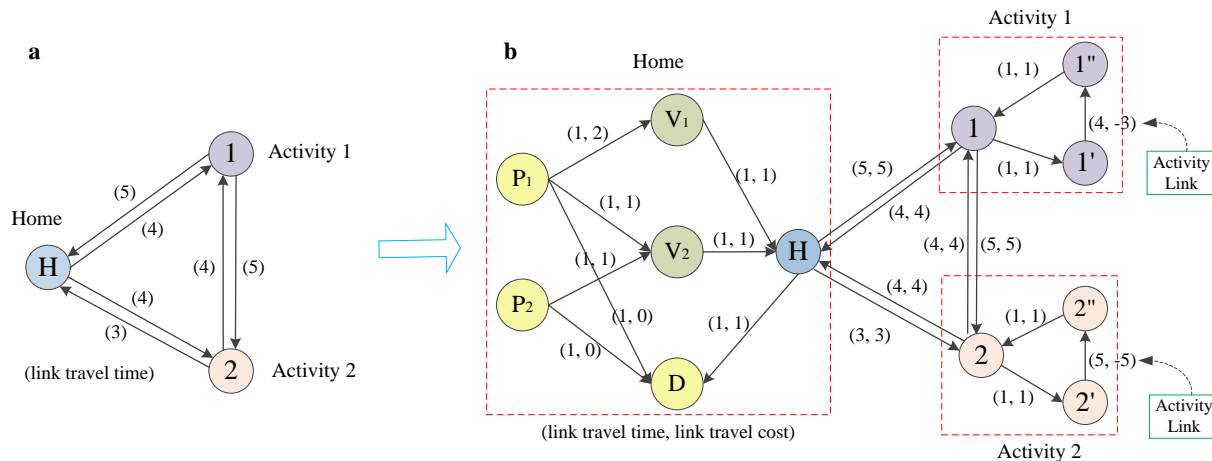
**Figure 7: Existing integration framework of ABM and DTA; (b) Proposed modelling framework of Case C.**

### 3.1.1 Network Construction and Conceptual Illustration of Case A

For illustrative purposes, a hypothetical three-node network shown in Fig. 8(a) is used to explain the problem addressed by Case A. There are two household members ( $p_1$  and  $p_2$ ), two available vehicles ( $v_1$  and  $v_2$ ), and two activities ( $a_1$  and  $a_2$ ). The available vehicle set of household members  $p_1$  and  $p_2$  is  $\{v_1, v_2\}$  and  $\{v_2\}$ . The available activity set of household members  $p_1$  and  $p_2$  is  $\{a_1, a_2\}$  and  $\{a_1\}$ , respectively, and the both activities belong to the mandatory activity and should be finished finally.

In order to model those requirements above, the physical network is modified as shown in Fig. 8(b), where the previous home node and activity nodes are split as several nodes. The detailed explanations are as follows. First, the home node is extended as six nodes, where (i) each household member has his/her dedicated node as his/her origin node, (ii) two vehicle nodes are created and one can view the links between household member origin node and vehicle nodes as vehicle selection links, while each vehicle node can only be visited less than or equal to once by

all passengers, and (iii) node D serves as the super destination node. Moreover, to follow an activity-on-the-link representation scheme, the extended network on the right has now activity starting node 1' and ending node 1'' corresponding to the activity node 1 on the left-hand side, and the link between the two nodes can be used to represent the required activity time duration.

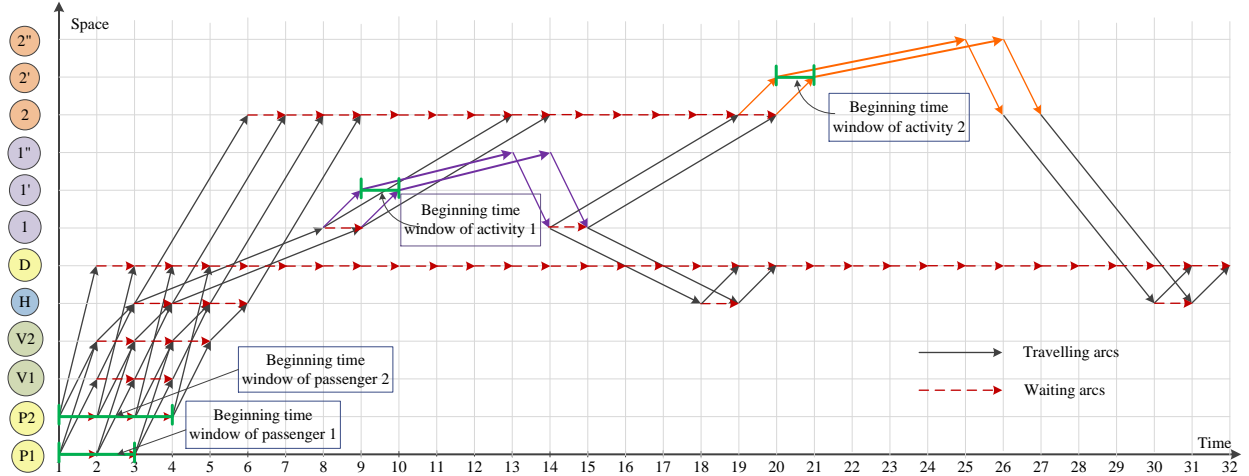


**Figure 8: (a) the physical network; (b) the corresponding modified network.**

In order to consider passenger-to-vehicle preference, the travel costs on those vehicle selection links can be passenger-specific. For example, the travel cost from passenger  $p_1$ 's origin to the super destination is 0, which indicates that when passenger  $p_1$  stays at home as one particular vehicle selection, there is no travel cost. In addition, the travel cost on link  $(p_1, v_1)$  is higher than that on link  $(p_1, v_2)$ , indicating passenger  $p_1$ 's higher preferences toward vehicle 2 compared to vehicle 1.

Each activity in HAPP typically has one specific time window, then we assume that the beginning time windows for (i) passengers  $p_1$  and  $p_2$  and (ii) activities  $a_1$  and  $a_2$  are  $[1, 3]$ ,  $[1, 4]$ ,  $[9, 10]$ , and  $[20, 21]$ , respectively, along the total time horizon of 32 time units. Furthermore, the waiting cost of each time interval at origin nodes and destination node is assumed to be 0, and the waiting at activities nodes has a cost of 1 at each time interval. Within a deterministic disutility minimization framework, we assume negative cost values on activity links shown in Fig. 2(b).

A standard time-discretized space-time network can be constructed through the procedure proposed in the papers (Tong et al., 2015; Li et al., 2015; Liu and Zhou, 2016; Lu et al., 2016), and the feasible space-time prism can be greatly reduced as illustrated in Fig. 9. As a result, the problem becomes how to find passengers' trajectory satisfying all time windows and activity requirements in the space-time network so as to minimize the total travel cost of all household members.



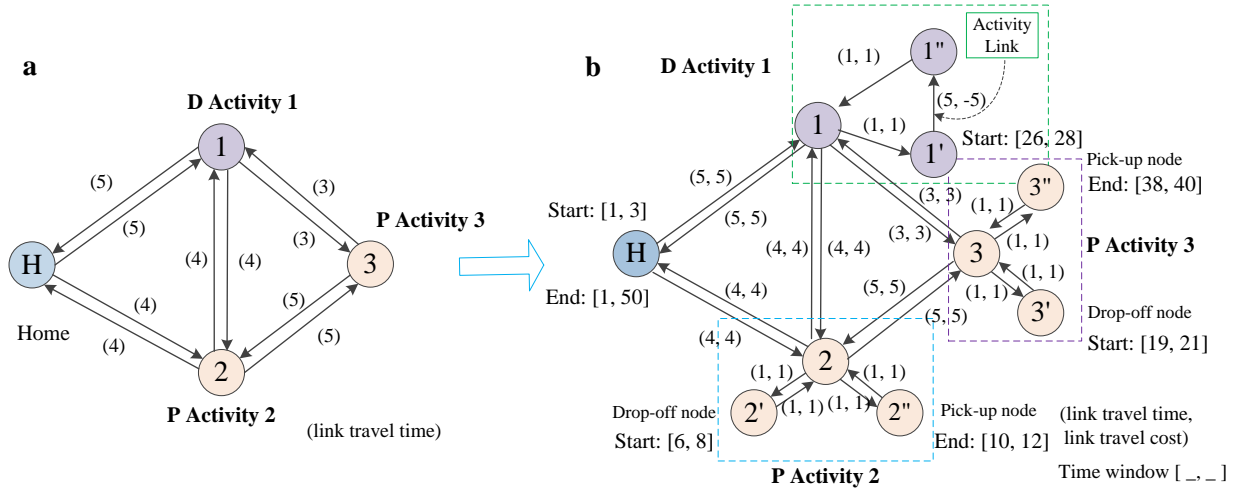
**Figure 9: Feasible searching region in the space-time network.**

As a remark, the (time-dependent) travel time on each travelling arc could be given in advance to reflect the congestion due to complex travel route choice interactions in the real-world traffic network. However, in the following Case C, we directly consider tight link/arc capacity inside the model to compute the resulting congestion effect explicitly. When the number of inflow vehicles exceeds the capacity of traveling arc, some vehicles have to wait at the waiting arc for available travelling arc capacity at next time interval. The detail about how tight capacity constraint is considered in the time-discretized space-time networks can be found in recent papers by Lu et al. (2016) and Liu and Zhou (2016). Their agent-based approach does not use the traditional flow-based nonlinear link/path cost function, and the travel cost of each agent is the result of the interaction among different agents in space-time networks.

### 3.1.2 Network Construction and Conceptual Illustration of Case A

The most difficult challenge in modeling the household-level ridesharing problem is how to recognize the complex coordination among different household members, pertaining to the following questions such as who is the driver and where/when the driver should drop off and pick up passengers. Considering offline planning applications, our Case B assumes that the set of possible ridesharing patterns is pre-specified with the given drop-off and pick-up locations with time windows to choose.

An illustrative example is given in Fig. 10(a), where there are two household members ( $p_1$  and  $p_2$ ), one available vehicle, and three activities ( $a_1$ ,  $a_2$  and  $a_3$ ). The given ride-sharing pattern requires that driver  $p_1$  needs to drop off the passenger  $p_2$  to his/her own activities within given beginning time windows, and then this driver needs to pick up  $p_2$  from the activity locations within given activity ending time windows. The driver  $p_1$  could accompany passengers to perform their activity, and also can leave to conduct his/her own mandatory activities. In this example with a quite busy household activity agenda, the driver has to perform the mandatory (driver as D) activity  $a_1$  while the passenger needs to finish the mandatory (passenger as P) activities  $a_2$  and  $a_3$ .



**Figure 10: (a) the physical network; (b) the corresponding modified network.**

Accordingly, we construct a drop-off node and a pick-up node for each passenger at the activity location in Fig. 10(b). It should be remarked that, in a typical case, one is dropped off and picked up at the same activity location, but our formulation also makes it possible that one passenger is dropped off at one activity and then picked up at another location if he/she can take other travel modes (walking, transit or taxi) to the (spatially different) pick-up location. The starting node and ending node of the passenger activity (shown as P activity 3' and 3'') is considered a special drop-off node and pick-up node in drivers' network.

In addition to using the two dimensions (space and time) to depict vehicles' travel trajectory, this section will introduce one more state dimension to model ride-sharing status. More precisely, the state code covers each traveler's service status, including the driver and all passengers. Through adding one more dimension and exogenously listing the possible relation of location, time, and vehicle state, a set of hard activity-performing constraints for the driver and passengers in each vehicle could be embedded in advance in the space-time-state network, which will greatly reduce the set of side constraints and make our proposed mathematical model tractable for network flow optimization algorithms.

To solve the single-vehicle routing problem with pickup and delivery service with time windows (VRPPDTW), Psaraftis (1983) proposed a cumulative service state  $\{1, 2, 3\}$  to record the service status of each passenger, where 3 means that the passenger has not been picked up, 2 means that the passenger has been picked up but not been delivered, and 1 means that the passenger has been successfully delivered. In this paper, we adopt the cumulative state representation as  $\{0, 1, 2\}$ : 0 means that the activity has not been performed, 1 means that the activity is being performed or the passenger has been dropped off at the activity location but not been picked up, and 2 means that the activity has been performed or the passenger has been picked up. While Mahmoudi and Zhou (2016) firstly proposed a third dimension as vehicle carrying state to solve the VRPPDTW, our third dimension of household-oriented state (with a rich representation of different household members, driver, passenger and associated activities) and the process of state transition are systematically different with those of Mahmoudi and Zhou (2016). The general comparison is listed in Table 3.

**Table 3: Comparison of model building between Mahmoudi and Zhou (2016) and our case B.**

|   | <b>Vehicle-oriented state based VRPPDTW (Mahmoudi and Zhou, 2016)</b>                                   | <b>Provide Output to Data Bus</b>   |
|---|---|---|
| State representation                                | Vehicle carrying state that indicates how many passengers are carried subject to its carrying capacity. | Vehicle, designated driver (as a household member) multiple passengers, activity-execution states                         |
| Multiple passenger activity task                    | Not modelled  | one passenger could conduct multiple mandatory and optional activities  |
| The third dimension (state)                         | 0: passenger is not carried by the vehicle;<br>1: passenger is being carried by the vehicle.            | 0: the activity of one passenger has not been performed;  |
| State transition logic between driver and passenger | Not modelled  | 1: the activity is being performed or the passenger has been dropped off at the activity location but not been picked up; |

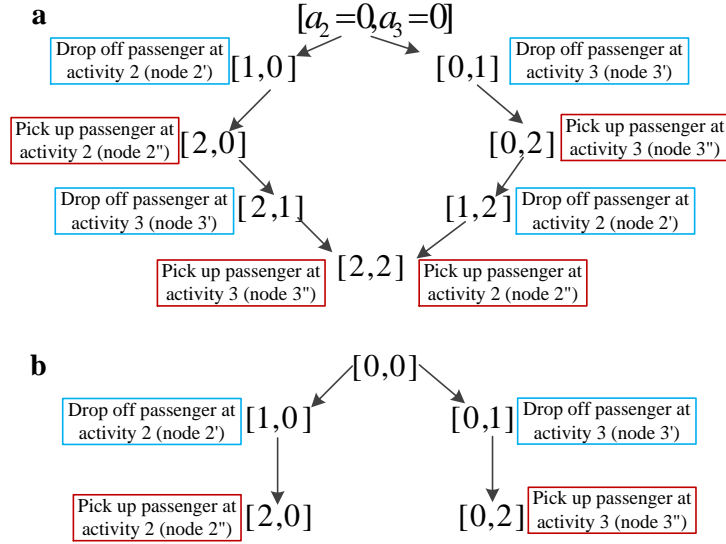
Since one activity could have 3 different states, if there are  $n$  activities for all passengers in one vehicle, it would require  $3^n$  variables to represent all possible states. The total number of states is shown in Table 2 depending on the number of activities. However, if one passenger has multiple activities, the possible states could be reduced because one passenger cannot perform multiple activities simultaneously. Also, the tight time window and transition preference for each activity can greatly reduce the number of possible states reasonable within a feasible space-time prism. In addition, the rapid development of hardware of computers could provide more memory and faster computation speed to address those large number of state search decisions.

**Table 4: The maximum number of possible states corresponds with the number of activities in one vehicle.**

| <b>Number of activities</b>       | 1 | 2 | 3  | 4  | 5   | 6   | 7    | 8    |
|-----------------------------------|---|---|----|----|-----|-----|------|------|
| Maximum number of possible states | 3 | 9 | 27 | 81 | 243 | 729 | 2187 | 6561 |

We now use the example above to illustrate our cumulative activity-performing state and the state transition at different activity locations and times. There are one vehicle with two household members and three activities, so the vehicle's activity-performing state can be  $[a_1, a_2, a_3]$ , or more generically denoted as  $[\_, \_, \_]$ , where the first slot represents the driver's activity-performing state of activity  $a_1$  and the second slot and the third one represent passenger  $p_2$ 's two activity-performing states of activities  $a_2$  and  $a_3$ , respectively. To reduce the number of states in this combinatorial optimization problem, one can also implement the activity-performing requirement as constraints on the activity link ( $1' \rightarrow 1''$ ) for the driver, so the resulting reduced state vector is  $[a_2, a_3]$ .

Since activities  $a_2$  and  $a_3$  are mandatory for passenger  $p_2$ , all possible vehicle's state could be  $[a_2 = 0, a_3 = 0]$ ,  $[1, 0]$ ,  $[2, 0]$ ,  $[0, 1]$ ,  $[0, 2]$ ,  $[2, 1]$ ,  $[1, 2]$ , and  $[2, 2]$  by enumeration. It is noticed that  $[a_2 = 1, a_3 = 1]$  is not included because it is impossible that passenger  $p_2$  is dropped off at two locations simultaneously. Fig. 11(a) illustrates a graph of possible state transitions for the example above. In addition, if there is the same type of multiple activities, such as, shopping at location 2 vs. at location 3, passenger 2 may just need to choose one of the two locations, so the resulting possible state transition will be that shown in Fig. 11(b).



**Figure 11: (a) both activities need to be performed; (b) exact one of two activities should be performed.**

There are three types of mutually exclusive multidimensional arcs in the space-time-state network:

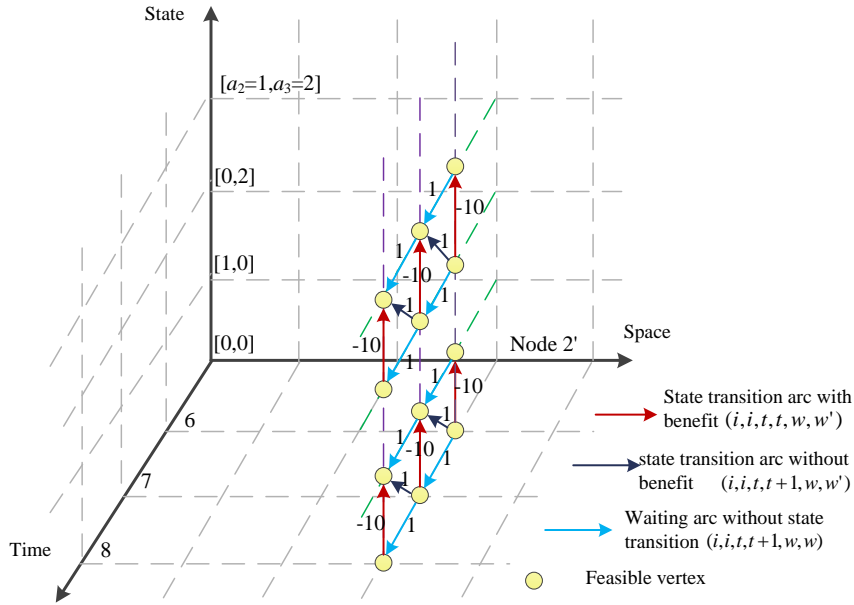
(1) Travelling arcs  $(i, j, t, s, w, w' = w)$  with a time-dependent cost on link  $(i, j)$  departing at time  $t$ , with the same state  $w$  as transportation services do not change activity performing states.

(2) Waiting arcs  $(i, i, t, t+1, w, w')$  with a unit of waiting costs at location  $i$  from time  $t$  to time  $t+1$ . A special Case is that, the waiting cost should be zero at the super home origin and destination nodes.

(3) State transition/service arcs  $(i, i, t, t, w, w')$  with a utility (i.e. negative travel cost) when performing their activities at the drop-off location. As shown in Fig. 6, at node  $i=2'$ , time  $t= 6, 7$  or 8 within a given time window, we have a number of possible state changes, for example,  $w=[0,0]$  with a possible transition to  $w'=[1,0]$ , or  $w=[0,2]$  with a possible transition to  $w'=[1,2]$ .

As the ending state for  $a_2$  must be 2 so the passenger 2 will be picked up automatically among any feasible solutions and there is no benefit at pick-up nodes to avoid double counting of service utilities.



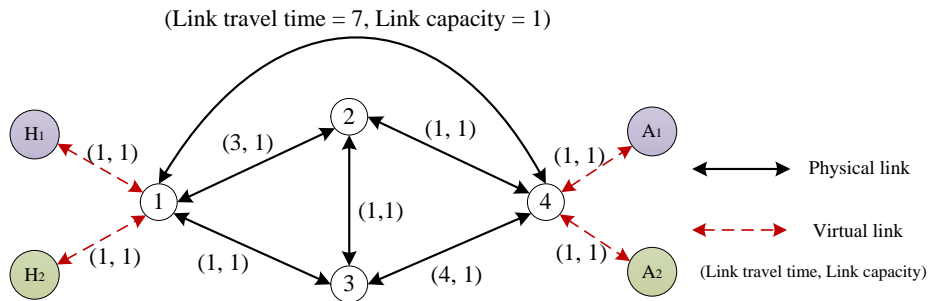


**Figure 12: Feasible arcs at node 2' in a space-time-state network.**

### 3.1.3 Conceptual Illustration of Case C

As an extension of Cases A and B, Case C strictly honors the travelling arc capacity in the space-time network and space-time-state network, similar to the consideration in the recent papers along this line (Lu et al., 2016; Liu and Zhou, 2016). Compared with the constant link free-flow travel time, the underlying time-varying congestion could dramatically affect the passenger/vehicle's departure time, route choice, mode choice, destination choice, and even activity generation.

Without loss of generality, we adopt a time-invariant network (Liu and Zhou, 2016) shown in Fig. 13 to illustrate the congestion effect for two households with two different activities, where household 1 (household 2) has one member who departs from home node  $H_1$  ( $H_2$ ) to perform activity  $A_1$  ( $A_2$ ) then go back home, respectively.



**Figure 13: Feasible arcs at node 2' in a space-time-state network.**

What can be observed in Table 5 is summarized as follows and those observation can also be applicable to space-time and space-time-state networks.

(1) When the link capacity is not taken into account, the vehicles from both households choose their own shortest path. The physical path node sequences of households 1 and 2 is  $1 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow 2 \rightarrow 3 \rightarrow 1$  with path travel time of 6.





(2) When the link capacity is considered, the system optimal objective of Case C could make household 1 change its path as  $1 \rightarrow 2 \rightarrow 4 \rightarrow 2 \rightarrow 1$  with a larger path cost of 8. Meanwhile, household 2 would switch a new path as  $1 \rightarrow 3 \rightarrow 4 \rightarrow 3 \rightarrow 1$  with an increased cost of 10.

(3) In observation (1), the travel time of household 2 from 1 to 4 is 3, but now it will increase to 5 due to link capacity constraint in observation (2). If the passenger of household 2 has a strict time window for activity 2, the increased path travel time from 1 to 4 could make passenger depart earlier to satisfy the time window.

(4) If the time budget of household 2 from 1 to 4 is less than 5, the passenger would cancel activity 2 or may change to an alternative by switching to other possible travel modes.

(5) If Case D is considered for possible equilibrium conditions, one household could choose the previous shortest path and the other has to accept the longer path,  $1 \rightarrow 4 \rightarrow 1$ , with total travel time of 14. It also could lead to changes in departure time, activity cancel or mode choice. In addition, Braess paradox exists in the network above, so blocking links  $3 \rightarrow 2$  and  $2 \rightarrow 3$  definitely could improve the transportation efficiency and further influence household activity patterns from the perspective of traffic managers.

**Table 5: Result analysis of different cases.**

| Different cases   |                                     | Physical path selection   |   |   |  | Remarks   |
|---|-------------------------------------|---|---|---|--|---|
|   |                                     | Path 1<br> | Path 2<br> | Path 3<br> | Path 4<br> |   |
| Case A/B: without link capacity constraint                  |                                     | ×   | ×   | ×   | √; √   | Benchmark   |
| Case C: system optimal with link capacity constraint        |                                     | √   | √   | ×   | ×  | Compared with Case A/B, household has possible departure time change, route choice change, and possible activity cancel or mode choice change due to link capacity constraints. |
| Case D: household equilibrium with link capacity constraint | With links $3 \leftrightarrow 2$    | ×   | ×   | √   | √  | Compared with Case C, the total travel time is increased.   |
|   | Without links $3 \leftrightarrow 2$ | √   | √   | ×   | ×  | Braess paradox occurs, as the system-wide cost reduces without the link.  |

√: One household (vehicle) chooses the corresponding path;

×: No household (vehicle) chooses the corresponding path;

## 3.2 MATHEMATICAL PROGRAMMING MODELS

### 3.2.1 Space-time Network-based Optimization Model for Case A

Based on the two-dimension space-time network constructed in section 2.2, we formulate our mathematical programming model that satisfies all requirements in Case A, which aims to optimize vehicle selection, activity-performing selection and route guidance for each household member so as to minimize the total household travel cost.

#### Model 1:

Objective function

$$\min \sum_p \sum_{(i,j,t,s) \in E} (c_{i,j,t,s}^p \times x_{i,j,t,s}^p) \quad (1)$$

Subject to

(1) Flow balance constraint for each person:

$$\sum_{i,t:(i,j,t,s) \in E} x_{i,j,t,s}^p - \sum_{i,t:(j,i,s,t) \in E} x_{j,i,s,t}^p = \begin{cases} -1 & j = O(p), s = DT(p) \\ 1 & j = D(p), s = T? \\ 0 & otherwise \end{cases} \quad \forall p \quad (2)$$

(2) Vehicle selection constraint at vehicle selection node:

$$\sum_p \sum_{i,t:(i,j,t,s) \in E} x_{i,j,t,s}^p \leq 1, \forall j \in N_v \quad (3)$$

(3) Mandatory activity participation for one specific household member:

$$\sum_{i,t:(i,j,t,s) \in E(p,a_m)} x_{i,j,t,s}^p = 1, \forall p \in P_m \quad (4)$$

(4) Mandatory activity with multiple candidates for one household member:

$$\sum_{a_n \in A_n(p)} \sum_{i,t:(i,j,t,s) \in E(p,a_n)} x_{i,j,t,s}^p = 1, \forall p \in P_n \quad (5)$$

(5) Discretionary activity for each household member

$$\sum_{i,t:(i,j,t,s) \in E(p,a_q)} x_{i,j,t,s}^p \leq 1, \forall p \in P_q \quad (6)$$

(6) Binary variable:  $x_{(i,j,t,s)}^p = \{0,1\}$

The objective function is to minimize the total system travel cost of all household members, where the travel cost  $c_{i,j,t,s}$  on each arc has been predefined in the space-time network construction stage. Eq. (2) is the standard person-based flow balance constraint. Eq. (3) means that each household member can only choose one vehicle or don't choose any vehicles (a.k.a. staying at home in our example). Eq. (4) represents that the activity duration arc of each mandatory activity of a specific household member should be visit exactly once by that household member. For example, if one household member must go to a company for work, one of working arcs must be visited exactly once by the household member. Eq. (5) ensures that if one household member needs to perform one type of activity with multiple candidate locations/time durations, he/she must choose one candidate to complete one activity instance among all options. For example, if one household member needs to go shopping and there are two candidate shopping malls, finally only one shopping mall should be visited exactly once to mark the completion state of the shopping activity. Inequality (6) represents the flexibility associated with those optional activities, as they could be performed or not, depending on the availability of those eligible household members and the required travel cost to reach those locations. In short, the proposed model in this section is a 0-1 integer linear programming model, or more precisely, a multi-commodity flow optimization problem with a limited set of side constraints. This compact formulation enables the use of standard optimization solvers for a real-world transportation network.

Table 6 offers a systematic comparison for detailed modelling techniques between our proposed model and classical model proposed by Recker (1995), specifically between our Case A and Case IV of Recker.

**Table 6: Comparison between Case IV (Recker, 1995) and our Case A.**

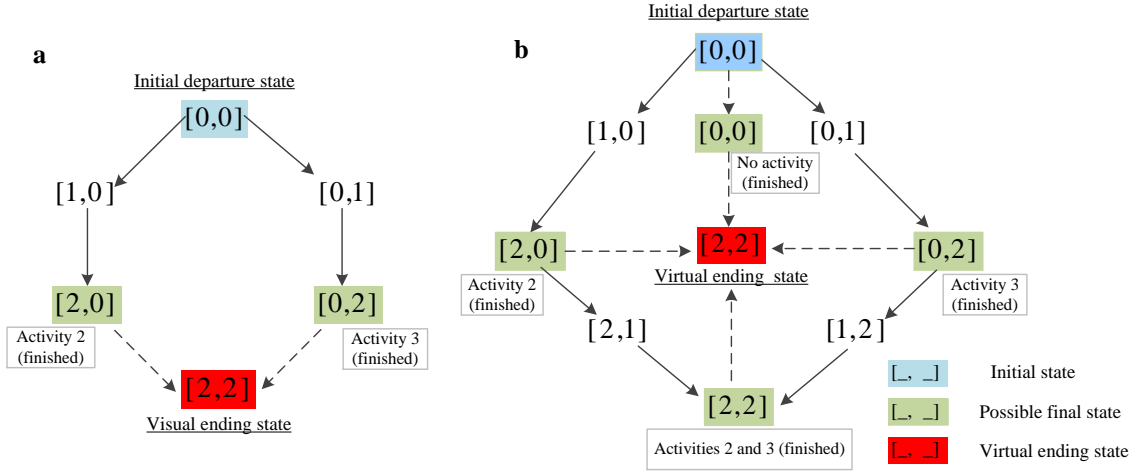
| Modelling constraints  | Model R4: Case IV (Recker, 1995)         | Model 1 for our Case A  | Remarks  |
|--|--|---|--|
| (1) Time representation  | Continuous                               | Discretized   |  |
| (2) Network representation   | Abstract physical traffic network        | Time-discretized space-time physical traffic network                            |  |
| (3) Objective function   | Eqns (1a)-(1f) with multiple goals       | Eqn (1) with travel cost only   |  |
| (4) Coupling constraints for vehicle selection of household member | Constraints (40a)-(40b)                  | Embedded in the modified physical network                                       |  |
| (5) Vehicle spatial connectivity constraints                       | Constraints (2), (3), (4'), (5') and (6) | Constraints (2)-(6) in the space-time network for modelling constraints (5)-(9) | Model 1 needs to build one specific activity duration link for each activity to represent the activity process |
| (6) Vehicle temporal constraints                                   | Constraints (7)-(10)                     |   |  |
| (7) Household spatial constraints                                  | Constraints (26)-(30)                    |   |  |
| (8) Household temporal constraints                                 | Constraints (31)-(33)                    |   |  |
| (9) Illogical activity constraints                                 | Constraints (21)-(24) and (36)-(39)      |   |  |

|   |                                     |   |   |
|---|-------------------------------------|---|---|
| (10) Vehicle capacity constraints       | Constraints (14)-(17)               | Always satisfied (solo driving pattern) |   |
| (11) Activity time window constraints   | Constraints (11)-(13) and (34)-(35) | Embedded in the space-time network      | Model R4 provides a starting time window and return-home window for each activity, but in Model 1 each activity only has a starting time window and does not have the return-home window. Instead, each household member has a return-home window for his/her arrival at home |
| (12) Travel cost/time budget constraint | Constraints (18)-(19)               | Not considered but can be easily added  |   |
| (13) Variable definitional constraints  | Binary and continuous variables     | Binary variables only                   | Model R4 is a mixed integer linear programming model. Model 1 is a 0-1 integer linear programming model.  |

### 3.2.2 Space-time-state Network-based Optimization Model for Case B

Before presenting the model for case B, it should be emphasized that the space-time-state network needs to be pre-built and satisfies the given time windows of each activity and the predefined arc attributes, such as, the location of each node, the travel time or travel cost of each arc, and the logically feasible state transition in the three-dimension network. More importantly, the slate of passengers' activity-performing states in the final solution for each vehicle exactly depends on the type of different activities, mandatory activity vs. discretionary activity. As shown in Fig. 11(a) in section 3.1.2, when the two activities are mandatory for passenger 2, the super starting state at the origin and super ending state at the destination are ["0,0"] and ["2,2"], respectively. On the other hand, when only one of two activities needs to be executed in a daily schedule in Fig. 11(b), the final arrival state could be ["2,0"] or ["0,2"], with a virtual ending state shown in Fig. 14(a).

Similarly, if the two activities are optional, the final state could be one of four possible alternatives ["0,0"], ["2,0"], ["0,2"] or ["2,2"], while the final selection of the optimal activity states is highly depending on the vehicle and time resources it consumed along the daily activity chain as well as the corresponding objective function in terms of benefit and travel costs. To satisfy the flow balance constraint for a network flow programming model, we need to build a virtual super ending state, as shown in Fig. 14(b), with connections from those possible ending states at the physical destination, e.g., four states in the above example, ["0,0"], ["2,0"], ["0,2"] and ["2,2"]. As a remark, there is no benefit/utility during the state transition to the virtual super ending state. This state-transition based modeling paradigm could systematically capture the complicated possible interactions between multiple household members in a daily scheduling process.



**Figure 14: (a) one of two activities should be performed; (b) two activities are optional.**

Based on the prebuilt 3D space-time-state network and given ride-sharing patterns, we now present our optimization model that satisfies all requirements in Case B that provides the optimal vehicle route guidance to the driver(s) to enable the scheduling of everyone's activities.

## Model 2:

Objective function

$$\min \sum_v \sum_{(i,j,t,s,w,w') \in E} (c_{i,j,t,s,w,w'}^v \times x_{i,j,t,s,w,w'}^v) \quad (7)$$

Subject to,

(1) Flow balance constraint for each vehicle:

$$\sum_{i,t,w:(i,j,t,s,w,w') \in E} x_{i,j,t,s,w,w'}^v - \sum_{i,t,w:(j,i,t,s,w,w') \in E} x_{i,j,t,s,w,w'}^v = \begin{cases} -1 & j = O(v), s = DT(v), w = [0, 0, \dots, 0] \\ 1 & j = D(v), s = T, w = [2, \dots, 2] \\ 0 & \text{otherwise} \end{cases}, \forall v \quad (8)$$

(2) Mandatory activity performing constraint for the driver on the activity arcs (including ride-sharing):

$$\sum_{i,t,w:(i,j,t,s,w,w') \in E(v,a_m)} x_{i,j,t,s,w,w'}^v = 1, \forall a \in A(v) \quad (9)$$

(3) Binary variable:  $x_{i,j,t,s,w,w'}^v = \{0, 1\}$

The objective function in Eq. (7) aims to minimize the total travel cost of the household, including the travel cost of vehicles and the benefit from everyone's performed activities. Eq. (8) is the

standard vehicle-based flow balance constraint. With the given initial departure state  $[0,0, \dots, 0]$  and virtual ending states for each vehicle, the given activity requirements of each passenger have been embedded in the space-time-state network. Similar to Eq. (4), Eq. (9) ensures that the household driver can finish his/her mandatory activity with given time windows and time duration, which means that the activity duration arc of each mandatory activity should be visited exactly once by the driver/vehicle. The decision variable  $x_{i,j,t,s,w,w'}^v$  is a binary variable that indicates whether or not the arc  $(i, j, t, s, w, w')$  will be chosen in the space-time-activity path of vehicle  $v$ . Finally, the model we proposed is also a 0-1 integer linear programming model, which has one more dimension compared to Case A but still can be directly solved in GAMS in a reasonable-size network.

In our Case B, the ridesharing pattern is prescribed, so the case can be viewed as a sub-problem of Case V in the paper (Recker, 1995). In Case V (Recker, 1995), it requires to build drop-off and pick-up nodes at each activity location and the set of available vehicles is expanded by designating driver seat and passenger seat(s) for each vehicle. The corresponding model has six categories of constraints, including vehicle temporal constraints, household member temporal constraints, vehicle spatial constraints, household member spatial constraints, vehicle capacity and budget constraints, and vehicle and household member coupling constraints. In our Case B, we also build drop-off and pick-up nodes for each activity with specific time windows. Since the ridesharing pattern is given a priori and modelled as a pair of drop-off-first then pick-up actions, we do not need to identify the specific driver seat and passenger seat(s), and the coupling constraints for vehicle and household member is automatically coded through the state transition graph or explicitly taken as activity-performing constraints in Eq. (9). The temporal and spatial constraints of vehicle and household member are all embedded in the well-structured space-time-state network where the state transition graph defines the possible activity visit sequences of passengers/vehicles.

It should be reminded that if we treat the start node and the end node of the activity duration link for the driver as a drop-off node and pick-up node, respectively, the driver's activities can also be added into the cumulative activity-performing state. As a result, side constraints (9) can also be embedded in the space-time-state network, and the mathematical model above is reduced to be a time-dependent state-dependent least cost path-finding problem, which could be efficiently solved by dynamic programming with parallel computing technology on large-scale networks. One multi-loop label correcting algorithm is designed in Appendix A.

As a remark, it is also possible to define another state instead of cumulative activity-performing state to model Case B. Based on the specific requirements in one problem, different state definitions could lead to different model formulations (less or more side constraints), different network structure and computation complexity. One specific example can be found in recent papers by Mahmoudi and Zhou (2016) and Mahmoudi et al., (2016) where they applied vehicle carrying state  $\{0,1\}$  and vehicle cumulative service state  $\{0,1,2\}$  to solve the VRPPDTWs, respectively, with different model formulation, networks, and algorithms. Therefore, our proposed formulation for Cases A and B is not the only possible modelling choice, and one should examine the size of state variables and nature of complex constraints to reformulate the problem based on the preferred network structure and available space and time complexity requirements.

### 3.2.3 Link Capacity Constraints of Case C

Since Case A considers a solo-driving pattern, one vehicle can only carry one person. In the mathematical model of Section 3.2.1, the person-based formulation is equivalent to the vehicle-based model. After converting the hourly road capacity into "specific time interval-based travelling arc capacity in the space-time network, the tight arc capacity constraint can be formulated as,

$$\sum_p x_{i,j,t,s}^p \leq cap_{i,j,t,s}, \forall (i, j, t, s) \in E \quad (10)$$

To consider the "queue spillback" phenomenon, additional inequality needs be added to represent the link storage capacity constraint by using cumulative arrival counts and cumulative departure counts on that link. The detailed formulation can be found in the paper by Li et al. (2015).

Similarly, since the mathematical model of Section 3.2.2 is vehicle-based formulation, the tight capacity constraint can be formulated as,

$$\sum_v x_{i,j,t,s,w,w'}^v \leq cap_{i,j,t,s,w,w'}, \forall (i, j, t, s, w, w') \in E \quad (11)$$

Regarding the queue spillback and congestion propagation property from Newell's simplified Kinematic wave model, the specific formulation is similar to the constraints in the paper by Li et al. (2015) which doesn't consider the merge and diverge issues, but with one more dimension  $w$ .

As stated at the end of Section 3.2.2, the driver's activity participation constraint can also be embedded in the space-time-state networks so that case B becomes a time-dependent state-dependent least cost path-finding problem. When the road resource capacity constraint (11) is recognized, there are two research directions to solve our proposed system optimal problem for large-scale real-world applications:

(1) Lagrangian relaxation: the link/arc capacity constraints can be dualized to objective function (7), so a new time-dependent state-dependent least cost path problem is transformed in the Lagrangian relaxation framework to obtain a lower bound. Since the optimal sub-gradient in binary integer programming model is hard to be obtained, the gap between the lower bound and the optimal solution cannot be well analytically proved. Meanwhile, when more side constraints from queue spillback consideration are taken into account, dualizing those constraints might not be a suitable approach.

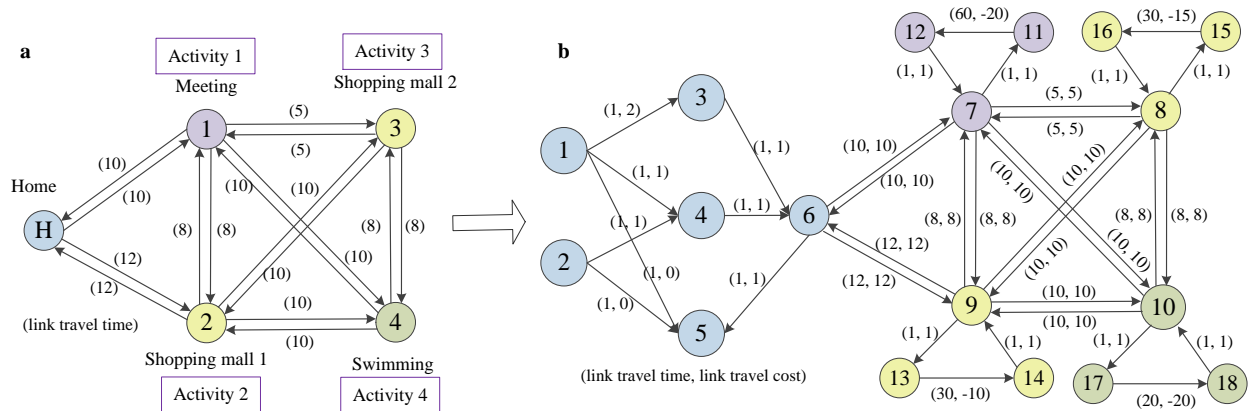
(2) Queue-based simulation: Since our proposed model is a system optimal problem considering complex traffic dynamics, we can apply event-based simulation to solve the large-scale problem where (i) the event-based simulation process is consistent with the time-discretized space-time-state network, (ii) different travel flow models can be handled, and (iii) the marginal cost analysis (Ghali and Smith, 1995) can be used to find the least marginal cost path for system optimal solutions. The specific algorithm design can refer to the paper by Lu et al (2016), which proposed a simulation framework to solve agent-based eco-system optimal traffic assignment in congested networks.



## 4.0 NUMERICAL EXPERIMENTS

### 4.1. Small-scale experiment for Case A

The proposed model for Case A in Section 3.1 will be tested in the following network shown in Fig. 9(a), where there are two household members  $p_1$  and  $p_2$ , two available vehicles  $v_1$  and  $v_2$ , and four candidate activities  $a_1$ ,  $a_2$ ,  $a_3$  and  $a_4$ . Household member  $p_1$  can choose any one of the two vehicles, and has one mandatory activity  $a_1$  to meet with others and one optional activity to swim. Household member  $p_2$  can only choose  $v_2$  and will go to one of the two shopping malls. The corresponding modified network is constructed in Fig. 9(b) where nodes 1 and 2 are origin nodes, nodes 3 and 4 are vehicle nodes, and node 5 is the final destination node. It is observed from the activity links that the time durations and costs for performing activities 1 to 4 are (60, -20), (30, -10), (30, -15), and (20, -20), respectively. The specific time windows are listed in Table 4. The waiting cost at each time interval is 0 at origin and destination nodes and 1 at activity nodes.



**Fig. 15 (a) the physical network; (b) the corresponding modified network**

**Table 7. The specific time window for each event**

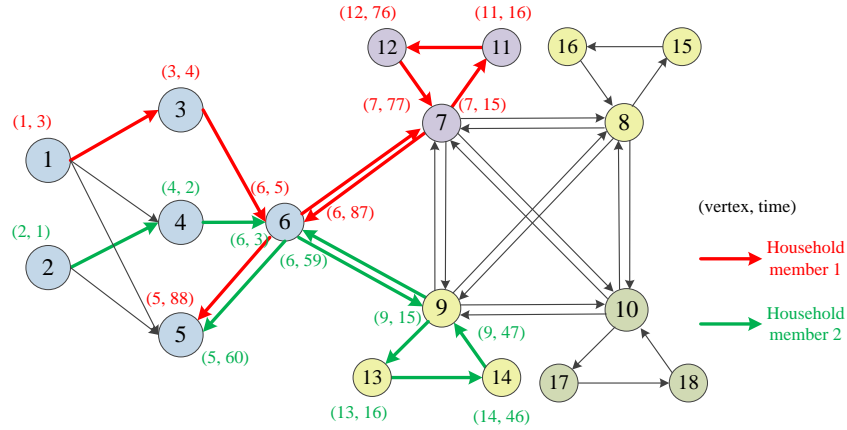
| Location (node) | 1      | 2      | 5        | 11       | 13       | 15       | 17       |
|-----------------|--------|--------|----------|----------|----------|----------|----------|
| Time window     | [1, 3] | [1, 3] | [1, 130] | [15, 18] | [15, 18] | [18, 20] | [86, 90] |

Our proposed 0-1 integer linear programming model for this example is solved in GAMS. The related source code can be downloaded at the website:

[https://www.researchgate.net/publication/306459026\\_Experiment\\_1\\_1](https://www.researchgate.net/publication/306459026_Experiment_1_1). Finally, the total travel cost of this household is 24. The specific optimal solution is listed in Table 5, and can be also illustrated in Fig. 10.

**Table 8. The optimal solution for each household member**

| Household member $p_1: x_{i,j,t,s}^1 = 1$ |     |     |     | Household member $p_2: x_{i,j,t,s}^2 = 1$ |     |     |     |
|---|-----|-----|-----|---|-----|-----|-----|
| $i$                                       | $j$ | $t$ | $s$ | $i$                                       | $j$ | $t$ | $s$ |
| 1   | 3   | 3   | 4   | 2   | 4   | 1   | 2   |
| 3   | 6   | 4   | 5   | 4   | 6   | 2   | 3   |
| 6   | 7   | 5   | 15  | 6   | 9   | 3   | 15  |
| 7   | 11  | 15  | 16  | 9   | 13  | 15  | 16  |
| 11  | 12  | 16  | 76  | 13  | 14  | 16  | 46  |
| 12  | 7   | 76  | 77  | 14  | 9   | 46  | 47  |
| 7   | 6   | 77  | 87  | 9   | 6   | 47  | 59  |
| 6   | 5   | 87  | 88  | 6   | 5   | 59  | 60  |



**Fig. 16 The trajectories of two household members**

It is observed that  $p_1$  should not go to activity 4 (swimming) and  $p_2$  does not need to go to activity 3 (shopping mall 2) due to the trade-off between the required travel costs and corresponding activity benefits. Therefore, if we increase the benefits of activities 3 and 4 to 17 and 23, respectively, the optimal solution will be that (i) the total cost is 22, (ii) household member  $p_1$  will visit activities 1 and 4 sequentially and then go back home, and (iii) household member  $p_2$  will visit activity 3 (shopping mall 2) rather than activity 2. Meanwhile, if we assume that the link travel time increases due to tight link capacity constraints when more other household activity trips are considered, the activity pattern of this household is expected to change again. In short, the final activity selection and route guidance are comprehensively evaluated and selected based on the possible time-varying travel cost in the physical network, available time windows, and the benefits of performing individual available activities.

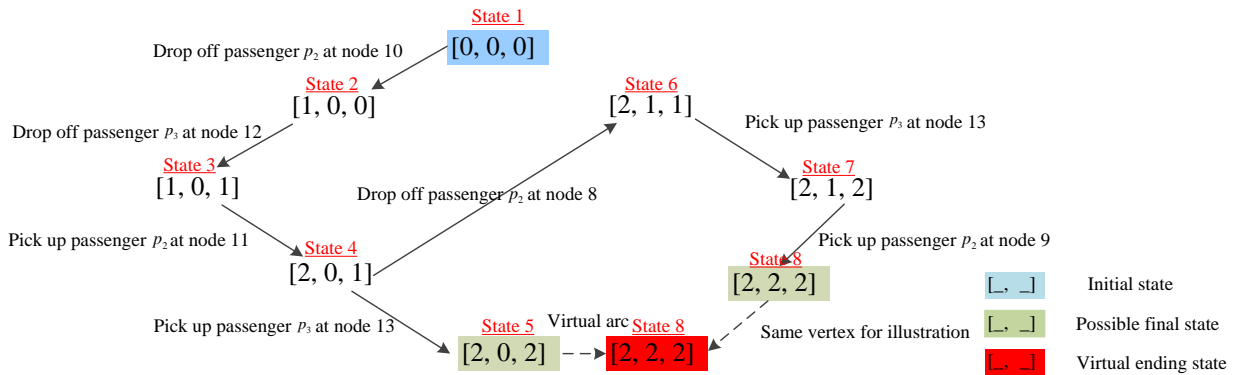
#### 4.2. Small-scale experiment for Case B

This section will test our proposed model for Case B in Section 3.2 based on the network shown in Fig. 11(a), where there are three household members with one driver and two passengers. They will share one vehicle to perform their daily activities. The driver  $p_1$  has one mandatory activity  $a_1$  and needs to drop off and pick up two passengers to conduct their activities. Passenger  $p_2$  has



| Location (node number) | Node 1 (departure) | Node 1 (arrival) | Node 10  | Node 11    | Node 12  | Node 13    | Node 8     | Node 9     | Node 6   |
|------------------------|--------------------|------------------|----------|------------|----------|------------|------------|------------|----------|
| Time window            | [1, 3]             | [1, 170]         | [15, 16] | [114, 115] | [28, 30] | [127, 139] | [127, 129] | [137, 139] | [41, 43] |

Based on the time window information, it is impossible that (i) activity 3 happens before activity 2, (ii) the drop-off event and pick-up event of activity 4 happens before those of activity 2, respectively, and (iii) the drop-off event and pick-up event of activity 3 happens before those of activity 4. Therefore, the remaining possible states will be  $[0, 0, 0]$ ,  $[1, 0, 0]$ ,  $[1, 0, 1]$ ,  $[2, 0, 1]$ ,  $[2, 0, 2]$ ,  $[2, 1, 1]$ ,  $[2, 1, 2]$ , and  $[2, 2, 2]$ . For the convenience of implementation in algorithms, we can label each state with one corresponding ID, such as, using 1 to 8 to represent the eight states above sequentially. The final possible state transition is demonstrated in Fig. 12, where virtual arcs with virtual ending state are also built for developing a single-origin-to-single-destination problem. In short, the possible state transition can be reduced with consideration of time windows before constructing a space-time-state network. Meanwhile, it is also feasible and straightforward to list all possible state transition without involving time windows and then consider the relation among activity locations, time windows, and state transition to construct the space-time-state network directly. As a result, those impossible state transition can also be eliminated when solving our mathematical model, but the price of this method is to require a bigger computer memory to store all possible state transition and a more complex space-time-state network.



**Fig. 18 The possible state transition graph**

In addition, the benefit or negative cost of performing activity for all passengers is assumed to occur during the state transition at drop-off nodes, as illustrated in Section 2.3. The negative travel costs for activities 1, 2, 3, and 4 are given as -20, -10, -15, and -15. Based on the constructed space-time-state network, our proposed 0-1 integer linear programming model for this example is solved in GAMS. The related source code can be downloaded at the website: [https://www.researchgate.net/publication/306458887\\_Experiment\\_2\\_1](https://www.researchgate.net/publication/306458887_Experiment_2_1). Finally, the total travel cost of this household is 40. The specific optimal solution is listed in Table 8.

**Table 11. The optimal solution for the household**

| The only vehicle: $x_{i,j,t,s,w,w'}^1 = 1$ |     |     |     |     |      |   |     |     |     |     |     |      |   |
|--|-----|-----|-----|-----|------|---|-----|-----|-----|-----|-----|------|---|
| $i$  | $j$ | $t$ | $s$ | $w$ | $w'$ | Remarks   | $i$ | $j$ | $t$ | $s$ | $w$ | $w'$ | Remarks   |
| 1  | 4   | 3   | 15  | 1   | 1    | Depart at home at time 3  | 7   | 2   | 103 | 104 | 3   | 3    |   |
| 4  | 10  | 15  | 16  | 1   | 1    |   | 2   | 4   | 104 | 112 | 3   | 3    |   |
| 10   | 10  | 16  | 16  | 1   | 2    | State transition (passenger $p_2$ is dropped off at node 10 for activity 2) | 4   | 4   | 112 | 113 | 3   | 3    |   |
| 10   | 4   | 16  | 17  | 2   | 2    |   | 4   | 11  | 113 | 114 | 3   | 3    |   |
| 4  | 5   | 17  | 27  | 2   | 2    |   | 11  | 11  | 114 | 114 | 3   | 4    | State transition (passenger $p_2$ is picked up at node 11 for activity 2)   |
| 5  | 12  | 27  | 28  | 2   | 2    |   | 11  | 4   | 114 | 115 | 4   | 4    |   |
| 12   | 12  | 28  | 28  | 2   | 3    | State transition (passenger $p_3$ is dropped off at node 12 for activity 4) | 4   | 5   | 115 | 125 | 4   | 4    |   |
| 12   | 5   | 28  | 29  | 3   | 3    |   | 5   | 5   | 125 | 126 | 4   | 4    |   |
| 5  | 2   | 29  | 39  | 3   | 3    |   | 5   | 13  | 126 | 127 | 4   | 4    |   |
| 2  | 2   | 39  | 40  | 3   | 3    |   | 13  | 13  | 127 | 127 | 4   | 5    | State transition (passenger $p_3$ is picked up at node 13 for activity 4)   |
| 2  | 2   | 40  | 41  | 3   | 3    | 13  | 5   | 127 | 128 | 5   | 5   |      |   |
| 2  | 6   | 41  | 42  | 3   | 3    |   | 5   | 2   | 128 | 138 | 5   | 5    |   |
| 6  | 6   | 42  | 43  | 3   | 3    |   | 2   | 1   | 138 | 148 | 5   | 5    | Arrive at home at time 148<br>State transition (from final state to assumed final state, the virtual arc cost is 0) |
| 6  | 7   | 43  | 103 | 3   | 3    | The driver $p_1$ performs activity 1  | 1   | 1   | 148 | 149 | 5   | 8    |   |

It is observed that passenger  $p_2$  will not perform activity 3 due to the trade-off between the required travel costs and corresponding activity benefits. If we increase the benefit of activity 3 from 15 to 20, the optimal solution will change to be that (i) the total cost is 37, and (ii) activity 3 will be performed by passenger  $p_2$ . Moreover, when the link travel time is modelled as a time-dependent attribute due to road congestion effect, the final household activity pattern is expected to change accordingly.

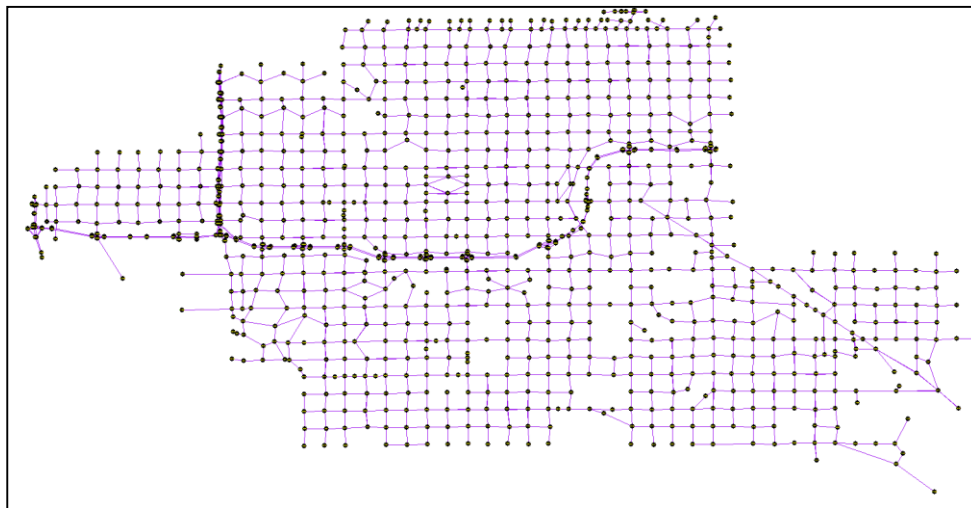
### 4.3. Medium-scale experiment within a Lagrangian relaxation framework using cumulative activity-performing state

This section aims to examine the computation efficiency of using cumulative activity-performing state for a general HAPP in a medium-scale transportation network. We choose a subarea of Phoenix regional network as our study case with 1186 nodes, 3164 links and 387 activity locations, shown in Fig. 13. The given input data for this experiment are listed in Table 9.

**Table 12. The input data of this experiment**

| agent_id  | agent_type | from_node_id | to_node_id | departure_time_start | departure_time_window | arrival_time_start | arrival_time_window | base_profit | optional |
|-----------|------------|--------------|------------|----------------------|-----------------------|--------------------|---------------------|-------------|----------|
| 1         | 0          | 23           | 23         | 30                   | 5                     | 40                 | 5                   | 150         | 0        |
| 2         | 0          | 24           | 24         | 10                   | 20                    | 70                 | 10                  | 133.33      | 0        |
| 3         | 0          | 26           | 26         | 40                   | 10                    | 60                 | 5                   | 83.33       | 0        |
| 4         | 0          | 25           | 25         | 20                   | 20                    | 80                 | 5                   | 150         | 0        |
| 5         | 0          | 39           | 39         | 70                   | 5                     | 90                 | 5                   | 133.33      | 0        |
| 6         | 0          | 35           | 35         | 20                   | 5                     | 110                | 5                   | 183.33      | 0        |
| 7         | 0          | 38           | 38         | 35                   | 10                    | 120                | 5                   | 133.33      | 1        |
| Vehicle 1 | 1          | 13           | 13         | 1                    | 1                     | 120                | 1                   |             |          |
| Vehicle 2 | 1          | 13           | 13         | 1                    | 1                     | 120                | 1                   |             |          |

“agent\_id” could be activity id or vehicle id. “agent\_type” = 1 for vehicles, and 0 means activities. Field “from\_node\_id” and “to\_node\_id” are the same and define (i) the activity performing location or (ii) vehicle’s origin/destination (home). “departure\_time\_start” defines the start time of activity or vehicle departure, and “departure\_time\_window” is the feasible time window duration. “arrival\_time\_start”, and “arrival\_time\_window” defines the activity/vehicle end time window. “base\_profit” is the benefit/utilities of performing the corresponding activity. The “optional” flag indicates that if an activity is optional, its value is 1, otherwise it is mandatory as 0. As a result, the problem becomes that two vehicles at home (node 13) plans to perform 6 mandatory activities and 1 optional activity.



**Fig. 19 One subarea of Phoenix regional transportation network**

To solve this problem, we use cumulative activity-performing state  $\{0,1,2\}$  to record the activity completion process. It is reminded that the maximum number of possible states could be  $3^7$  for 7 activities. In order to model the competition for one activity by two vehicles simultaneously, we dualize that constraint to our objective function and adopt the forward dynamic programming algorithm within a Lagrangin relaxation framework, which can refer to the process of solving the

VRPPDTW for multiple vehicles by Mahmoudi and Zhou (2016). The related C++ source code and data set can be downloaded at the website: <https://github.com/xzhou99/Agent-Plus/tree/master/HAPP>. Table 10 lists the impact of different number of activities on the CPU computation time of 5 Lagrangian iterations (for distributing different tasks to two vehicles) and computer memory usage. In the above case, the vehicle/activity preference for household members is not considered. If a pre-specified vehicle-to-activity mapping is given, the search space in the space-time-state network could be further reduced.

**Table 13. CPU computation time and memory use under different number of activities**

| # of activities | Maximum numbers of activity performing states | CPU time (seconds) | RAM (GB) |
|-----------------|---|--------------------|----------|
| 4               | 81  | 15.5               | 0.3      |
| 5               | 243   | 38.2               | 1.3      |
| 6               | 729   | 112.3              | 3.6      |
| 7               | 2187  | 337.4              | 11.3     |

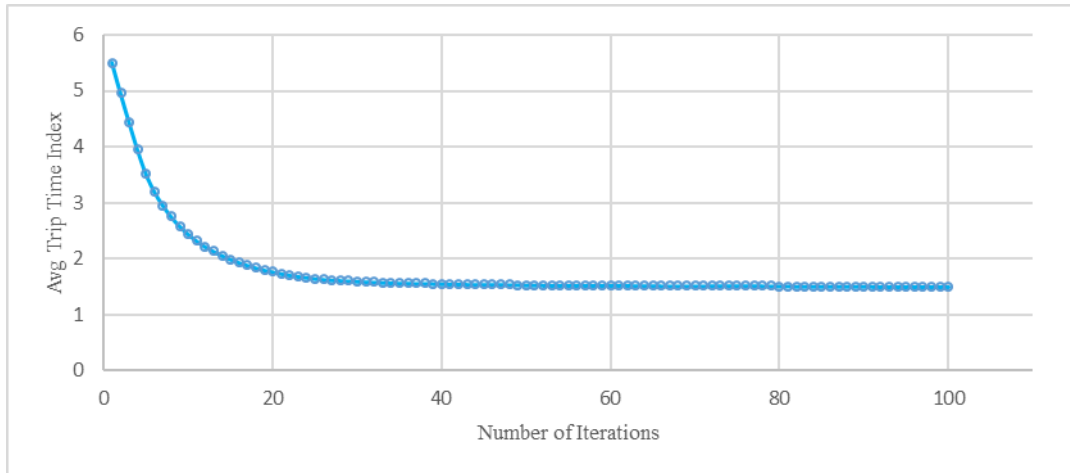
#### 4.4. Large-scale experiment within a simulation-based framework with simplified activity representation and road capacity constraints

This section aims to present the initial test result of the simulation-based approach for system optimal dynamic vehicle routing under road capacity constraints. The Salt Lake City regional traffic network is selected shown in Fig. 14 where there are 13,923 nodes, 26,768 links and 2,302 zones. The total number of simulated vehicles is about 1.35 million from 15:00 to 18:00. The traffic flow model chooses point queue model, which just considers the tight road capacity constraints. The details of implementing spatial queue model and Newell’s simplified kinematic wave model by simulation can be found in the paper (Zhou and Taylor, 2014).



**Fig. 20 Salt Lake City regional traffic network (Lu et al., 2016)**

This experiment can be treated as a special version of Case A. Each origin zone is analogous to one household and those destination zones can be viewed as those mandatory activity locations. The process that vehicles depart from origin to destination is like that household members complete their mandatory activities with flexible time windows. The simulated average trip time index (mean trip simulated travel time/trip free-flow travel time) of 100 iterations is depicted in Fig. 15 and finally shows a convergence pattern. A parallel computing technique (Qu and Zhou, 2017) is embedded in the simulation process, and the search process for single activity is extremely simple compared to the full scale space-time-state search presented in the medium-scale example, so the computational time for one iteration is just 1 min 25sec in our workstation with 40 available CPU threads and 192G memory. As stated in the paper (Lu et al., 2016), this simulation algorithm still needs further improvements on path marginal travel time calculation and step size optimization of each iteration.



**Fig. 21 Average trip time index of each iteration**

## 5.0 CONCLUSIONS

Despite all advancements in the real-time traffic control, DTA modelers still seek for a robust framework to extend their existing model (1) from single-OD demand to trip chaining, and (2) from driving your own mode to shared-use vehicle systems. In this research, a multi-resolution simulation platform is proposed to integrate ABM and DTA for better capture the interactions of travelers, activity, and traffic dynamics. In addition, by embedding a set of hard constraints into a well-structured (space-time and space-time-state) network structure, we reformulate two difficult cases in HAAP using a person-based or vehicle-based network flow programming model with very few side constraints. Meanwhile, the tight road capacity is highly considered for capturing the underlying congestion effect for the cases above. The numerical experiments demonstrate our



proposed methodology and analyze the impacts of different activity benefits on the final vehicle routing and household member activity selection.

## APPENDIX A

### MULTI-LOOP LABEL-CORRECTING ALGORITHM

```
 $L_{i,t,w}(v) := 0$  // label cost at vertex  $(i, t, w)$  for vehicle  $v$   
 $L_{j,s,w'}(v) := +\infty$ ; for each vertex  $(j, s, w') \in R - \{(i, t, w)\}$  // label cost at vertex  $(j, s, w')$  for vehicle  $v$   
node pred of vertex  $(.,.,.,.) := -1$ ;  
time pred of vertex  $(.,.,.,.) := -1$ ;  
state pred of vertex  $(.,.,.,.) := -1$ ;  
 $LIST := \{(i, t, w)\}$ ;  
While  $LIST \neq \emptyset$  do  
for each time  $t \in [0, T]$  do // adding time window of each activity can further reduce the searching region  
begin  
for each state  $w$  do // the number of states can be reduced by the activity sequence for each passenger  
begin  
for each link  $(i, j)$  do //  
begin  
derive downstream state  $w'$  based on the feasible state transition  
derive arrival time  $s = t + TT_{i,j,t}$ ;  
if  $(L_{i,t,w}(v) + c_{i,j,t,s,w,w'}(v) < L_{j,s,w'}(v))$   
begin  
 $L_{j,s,w'} := L_{i,t,w}(v) + c_{i,j,t,s,w,w'}(v)$  ; // label update  
node pred of vertex  $(v, j, s, w') := i$ ;  
time pred of vertex  $(v, j, s, w') := t$ ;  
state pred of vertex  $(v, j, s, w') := w$ ;  
if vertex  $(j, s, w') \notin LIST$  then add vertex  $(j, s, w')$  to  $LIST$   
end;  
end; // for each link  
end; // for each state  
end; // for each time
```

## REFERENCES

- Abdul Aziz, H.M., Ukkusuri, S., (2013). An Approach to Assess the Impact of Dynamic Congestion in Vehicle Routing Problems. In: *Advances in Dynamic Network Modeling in Complex Transportation Systems*. Springer US, 265-285.
- Arentze, T. A., Timmermans, H. J., (2004). A learning-based transportation oriented simulation system. *Transportation Research Part B* 38(7), 613–633
- Bhat, C. R., Guo, J. Y., Srinivasan, S., Sivakumar, A., (2004). Comprehensive econometric micro-simulator for daily activity-travel patterns. *Transportation Research Record: Journal of the Transportation Research Board* 1894(1), 57–66.
- Chow, J.Y., Djavadian, S., (2015). Activity-based market equilibrium for capacitated multimodal transport systems. *Transportation Research Part C* 59, 2-18.
- Chow, J. Y., Recker, W. W., (2012). Inverse optimization with endogenous arrival time constraints to calibrate the household activity pattern problem. *Transportation Research Part B* 46(3), 463–479.
- Cordeau, J.F., G. Desaulniers, J. Desrosiers, M.M. Solomon, F. Soumis. The VRP with time windows. P. Toth, D. Vigo, eds. *The Vehicle Routing Problem*, SIAM Monographs on Discrete Mathematics and Applications. SIAM, Philadelphia, PA, 2001, pp. 157–194.
- Fu, X., Lam, W.H., Xiong, Y., (2016). Modelling intra-household interactions in household's activity-travel scheduling behaviour. *Transportmetrica A: Transport Science* 12(7), 1-21.
- Gan, L.P., Recker, W., (2008). A mathematical programming formulation of the household activity rescheduling problem. *Transportation Research Part B* 42 (6), 571–606.
- Ghali, M.O., Smith, M.J., (1995). A model for the dynamic system optimum traffic assignment problem. *Transportation Research Part B* 29 (3), 155–170.
- Hu, X., Y.C. Chiu, S. Delgado, L. Zhu, R. Luo, P. Hoffer, and S. Byeon. Behavior insights for an incentive-based active demand management platform. In 93rd Annual Meeting of the Transportation Research Board, Washington, DC, 2014.
- Kang, J.E., Chow, J.Y., Recker, W.W., (2013). On activity-based network design problems. *Transportation Research Part B* 57:398-418
- Kang, J.E., Recker, W.W., (2013). The location selection problem for the household activity pattern problem. *Transportation Research Part B* 55, 75-97.
- Kitamura, R. An evaluation of activity-based travel analysis. *Transportation*, 1988, 15 (1-2), pp. 9-34.
- Li, P., Mirchandani, P., Zhou, X., (2015). Solving simultaneous route guidance and traffic signal optimization problem using space-phase-time hypernetwork. *Transportation Research Part B* 81(1), 103–130.
- Liao, F., Arentze, T., Timmermans, H., (2013a). Incorporating space–time constraints and activity-travel time profiles in a multi-state supernetwork approach to individual activity-travel scheduling. *Transportation Research Part B* 55, 41–58.

- Liao, F., Arentze, T., Timmermans, H., (2013b). Multi-state supernetwork framework for the two-person joint travel problem. *Transportation*, 40(4), 813-826.
- Lin, D.Y., Eluru, N., Waller, S., Bhat, C., (2008). Integration of activity-based modeling and dynamic traffic assignment. *Transportation Research Record: Journal of the Transportation Research Board* 2076, 52-61.
- Liu, J., Zhou, X., (2016). Capacitated transit service network design with boundedly rational agents. *Transportation Research Part B* 93, 225-250.
- Lu, C.C., Liu, J., Qu, Y., Peeta, S., Routhail, N.M., Zhou, X., (2016). Eco-system optimal time-dependent flow assignment in a congested network. *Transportation Research Part B* 94, 217–239.
- 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. *Transportation Research Part B* 89, 19-42.
- Mahmoudi, M., Chen, J., Zhou, X., (2016). Embedding Assignment Routing Constraints through Multi-Dimensional Network Construction for Solving the Multi Vehicle Routing Problem with Pickup and Delivery with Time Windows. arXiv:1607.01728.
- Miller, E. J., Roorda, M. J., (2003). Prototype model of household activity-travel scheduling. *Transportation Research Record: Journal of the Transportation Research Board* 1831 (1), 114–121.
- Morning Consult, 2015. National Tracking Poll #150505 – June 29–31, 2015. Morning Consult, Washington, D.C. [ONLINE] Available at: [https://morningconsult.com/wp-content/uploads/2015/06/150505\\_crosstabs\\_mc\\_v2\\_AD.pdf](https://morningconsult.com/wp-content/uploads/2015/06/150505_crosstabs_mc_v2_AD.pdf), Retrieved 15 July, 2016.
- Pendyala, R., Konduri, K., Chiu, Y.C., Hickman, M., Noh, H., Waddell, P., Wang, L., You, D., Gardner, B., (2012). Integrated land use-transport model system with dynamic time-dependent activity-travel microsimulation. *Transportation Research Record: Journal of the Transportation Research Board* 2303, 19-27.
- Pendyala, R., Kitamura, R., Kikuchi, A., Yamamoto, T., Fujii, S., (2005). Florida activity mobility simulator: overview and preliminary validation results. *Transportation Research Record: Journal of the Transportation Research Board* 1921(1), 123–130.
- Pendyala, R., You, D., Garikapati, V., Konduri, K., Zhou, X., (2017). Paradigms for integrated modeling of activity-travel demand and network dynamics in an era of dynamic mobility management. *The 96th Annual Meeting of the Transportation Research Board*, accepted for presentation only.
- Pribyl, O., Goulias, K.G., (2005). Simulation of daily activity patterns incorporating interactions within households: Algorithm overview and performance. *Transportation Research Record: Journal of the Transportation Research Board* 1926, 135-141.
- Psaraftis, H.N., (1983). An exact algorithm for the single-vehicle many-to-many dial-a-ride problem with time windows. *Transportation Science* 17 (3), 351–357.

- Qu, Y., Zhou, X., (2017). Large-scale dynamic transportation network simulation: a space-time-event parallel computing approach. *Transportation Research Part C* 75, 1-16.
- Rasouli, S., & Timmermans, H. Activity-based models of travel demand: promises, progress and prospects. *International Journal of Urban Sciences*, 2014, 18(1), pp. 31-60.
- Recker, W.W., (1995). The household activity pattern problem: general formulation and solution. *Transportation Research B* 29 (1), 61-77.
- Recker, W.W., (2001). A bridge between travel demand modelling and activity-based travel analysis. *Transportation Research Part B* 35(5), 481 – 506.
- Recker, W.W., Chen, C., McNally, M.G., (2001). Measuring the impact of efficient household travel decisions on potential travel time savings and accessibility gains. *Transportation Research Part A* 35 (4), 339-369.
- Sherman, D. Semi-Autonomous Cars Compared! Tesla Model S vs. BMW 750i, Infiniti Q50S, and Mercedes-Benz S65 AMG. Available at: <http://www.caranddriver.com/features/semi-autonomous-cars-compared-tesla-vs-bmw-mercedes-and-infiniti-feature>, 2016, Retrieved 15 July, 2016.
- Tong, L., Zhou, X., Miller, H.J., (2015). Transportation network design for maximizing space–time accessibility. *Transportation Research Part B* 81(2), 555–576.
- Tesla Motors Team. Your Autopilot Has Arrived. Available at: <https://www.tesla.com/blog/your-autopilot-has-arrived>, 2015, Retrieved 15 July, 2016.
- Zhou, X., Taylor, J., (2014). DTALite: a queue-based mesoscopic traffic simulator for fast model evaluation and calibration. *Cogent Engineering* 1 (1), 961345.