Mobility21
A USDOT NATIONAL
UNIVERSITY TRANSPORTATION CENTER

Carnegie Mellon University

Penn
University of Pennsylvania

The Ohio State University

CCAC

Mesoscopic car-truck flow modeling and simulation: theory and applications

Wei Ma, Xidong Pi, Zhen (Sean) Qian

Final Research Report

Contract # 69A3551747111

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the U.S. Department of Transportation's University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

Estimating multi-class dynamic origin-destination demand through a forward-backward algorithm on computational graphs

Wei Ma[a], Xidong Pi[a], Zhen (Sean) Qian[a,b,*]

[a]*Department of Civil and Environmental Engineering*
Carnegie Mellon University, Pittsburgh, PA 15213
[b]*H. John Heinz III Heinz College*
Carnegie Mellon University, Pittsburgh, PA 15213

Abstract

Transportation networks are unprecedentedly complex with heterogeneous vehicular flow. Conventionally, vehicle classes are considered by vehicle classifications (such as standard passenger cars and trucks). However, vehicle flow heterogeneity stems from many other aspects in general, e.g., ride-sourcing vehicles versus personal vehicles, human driven vehicles versus connected and automated vehicles. Provided with some observations of vehicular flow for each class in a large-scale transportation network, how to estimate the multi-class spatio-temporal vehicular flow, in terms of time-varying Origin-Destination (OD) demand and path/link flow, remains a big challenge. This paper presents a solution framework for multi-class dynamic OD demand estimation (MCDODE) in large-scale networks. The proposed framework is built on a computational graph with tensor representations of spatio-temporal flow and all intermediate features involved in the MCDODE formulation. A forward-backward algorithm is proposed to efficiently solve the MCDODE formulation on computational graphs. In addition, we propose a novel concept of tree-based cumulative curves to estimate the gradient of OD demand. A Growing Tree algorithm is developed to construct tree-based cumulative curves. The proposed framework is examined on a small network as well as a real-world large-scale network. The experiment results indicate that the proposed framework is compelling, satisfactory and computationally plausible.

1. Introduction

Transportation networks are unprecedentedly complex with heterogeneous vehicular flow. Conventionally, vehicle heterogeneity are considered by vehicle classifications (such as standard passenger cars and trucks). However, vehicle flow heterogeneity stems from many other aspects in general, e.g., ride-sourcing vehicles versus personal vehicles, human driven vehicles versus connected and automated vehicles. How to effectively manage the multi-class vehicles in a complex transportation system so as to improve the network efficiency presents a big challenge. As an indispensable component of dynamic transportation network models with heterogeneous traffic, the multi-class dynamic origin-destination (OD) demand plays a key role in transportation planning and management to understand spatio-temporal vehicular flow and its travel behavior. To our best knowledge, there is a lack of studies to understand and estimate the dynamic OD demand of multiple vehicle classes using sparse and partial flow observations. In view of this, this paper presents a data-driven framework for multi-class dynamic OD demand estimation (MCDODE) in large-scale networks. The MCDODE formulation is represented on a computational graph, and a novel forward-backward algorithm is proposed to estimate the OD demand efficiently and effectively. The proposed MCDODE framework is examined in a small networks as well as a real-world large-scale network to demonstrate the estimation accuracy and the computational efficiency.

The multi-class dynamic OD demand (MCDOD) represents the number of vehicles in each of the vehicle classes (*e.g.* personal cars, trucks, ride-sourcing vehicles) departing from an origin and heading to a destination in a particular time interval. The definition of "classes" is very general, by vehicle sizes, specifications, and

[*] Corresponding author
Email addresses: weima@cmu.edu (Wei Ma), xpi@cmu.edu (Xidong Pi), seanqian@cmu.edu (Zhen (Sean) Qian)

nature of trips. The MCDOD reveals the fine-gained traffic demand information for different vehicle classes and the overall spatio-temporal mobility patterns can be inferred from the OD demand and its resultant path/link flows. Policymakers can understand the departure/arrival patterns of multi-class vehicles through the MCDOD. The MCDOD also helps the policymakers understand the impact of each vehicle class on the roads, and hence each class of vehicles can be managed separately. In addition, most of Advanced Traveler Information Systems/Advanced Traffic Management Systems (ATIS/ATMS) would require accurate MCDOD as the model input (Huang & Li 2007).

The dynamic OD estimation (DODE) has been extensively studied over the past few decades. A generalized least square (GLS) formulation is proposed for estimating dynamic OD demand with exogenous route choice model (Cascetta et al. 1993). The GLS formulation is further extended to a bi-level optimization problem, and the bi-level formulation solves for the DODE with endogenous route choice model (Tavana 2001). Advantages and disadvantages of the bi-level formulation are discussed in Nguyen (1977), LeBlanc & Farhangian (1982), Fisk (1989), Yang et al. (1992), Florian & Chen (1995), Jha et al. (2004). The Stochastic Perturbation Simultaneous Approximation (SPSA) methods have also been adopted to solve the bi-level formulation in many studies (Balakrishna et al. 2008, Cipriani et al. 2011, Lee & Ozbay 2009, Vaze et al. 2009, Ben-Akiva et al. 2012, Lu et al. 2015, Tympakianaki et al. 2015, Antoniou, Azevedo, Lu, Pereira & Ben-Akiva 2015). Zhang et al. (2017), Osorio (2019) proposed an instantaneous approximation for the dynamic traffic assignment models, and the DODE problem is cast into a convex optimization which can be solved efficiently. The bi-level formulation can also be relaxed to a single-level problem and solved by updating the OD demand using gradient-based methods (Nie & Zhang 2008, Lu et al. 2013).

The DODE problem can be viewed as a statistical estimation problem. For example, a statistical inference framework using Markov Chain Monte Carlo algorithm was proposed to estimate the probabilistic OD demand (Hazelton 2008). Zhou et al. (2003) proposed a DODE framework with multi-day data and established a hypothesis testing framework to identify the demand evolution within a week. In addition to the off-line DODE methods, the DODE formulation can also be solved with real-time data streams for ATIS/ATMS applications. Bierlaire & Crittin (2004) extended the GLS formulation on large-scale networks and solved the formulation efficiently on a real-time basis. The state-space models are also used to estimate the dynamic OD of each time interval sequentially (Zhou & Mahmassani 2007, Ashok & Ben-Akiva 2000).

The DODE methods often encapsulate the dynamic traffic assignment (DTA) models within the bi-level formulation. Various DTA models can be adopted to model the network dynamics and travelers' behaviors. For example, Dynamic Network Loading (DNL) models (Ma et al. 2008) simulate the vehicle trajectories and traffic conditions given determined the origin-destination (O-D) demand and fixed routes; Dynamic User Equilibrium (DUE) models (Mahmassani & Herman 1984, Nie & Zhang 2010) search for the equilibrated travelers' route choices to achieve user optima; Dynamic System Optimal (DSO) models (Shen et al. 2007, Qian et al. 2012, Ma et al. 2014) explore the optimal conditions under which the total network costs are minimized.

The multi-class traffic assignment models and OD estimation models can be built by extending the single-class models. However, the main challenges are how to estimate MCDOD in such a way to match multi-source spatio-temporal data in a large-scale transportation network. A number of studies (Gundaliya et al. 2008, Venkatesan et al. 2008, Qian et al. 2017) investigate the multi-class traffic flows in general networks. The multi-class DTA models are studied by Dafermos (1972), Yang & Huang (2004), Huang & Li (2007) without the real-world data validation. There are studies estimating the static OD demand for multiple vehicle classes (Wong et al. 2005, Raothanachonkun et al. 2006, Noriega & Florian 2007, Zhao et al. 2018). In particular, Shao et al. (2015) estimated the probabilistic distribution of the multi-class OD using traffic counts. The research gap lies in estimating MCDOD in large-scale networks that are consistent with real-world multi-source traffic data.

In this paper, we develop a data-driven framework that estimates the multi-class dynamic OD demand using traffic counts and travel time data that are partially observed in general networks. The proposed framework formulates the MCDODE problem and represents it with a computational graph. The MCDODE is solved with a novel forward-backward algorithm on the computational graph. The MCDODE framework can be solved using multi-core CPUs or Graphics Processing Units (GPUs), and hence the proposed method can be computational efficient and applied to the large-scale networks and multi-day data. The closest work to this paper is that of Wu et al. (2018), which constructs a layered computational graph for the single-class static OD estimation problem. This paper extends the computational graph approach to the case of multi-class dynamic OD demand by tackling additional challenges on the incorporating flow dynamics and characteristics across both classes and time of day.

We also build a novel framework to evaluate the gradients of multi-class OD demand for simulation-based traffic assignment models. The main contributions of this paper are summarized as follows:

1) We propose a theoretical formulation for estimating multi-class dynamic OD demand. The formulation is represented on a computational graph such that the MCDODE can be solved for large-scale networks with large-scale traffic data. The proposed MCDODE formulation can handle any form of traffic data, such as flow, speed or trip cost.

2) We propose a novel forward-backward algorithm to solve for the MCDODE formulation on the constructed computational graph with simulation-based traffic assignment models.

3) We use a tree-based cumulative curves to evaluate the gradient of multi-class dynamic OD demand in the forward-backward algorithm, and a Growing Tree algorithm is proposed to construct the tree-based cumulative curves during the traffic simulation.

4) We examine the proposed MCDODE framework on a large-scale network to demonstrate its effectiveness and computational efficiency of the solution algorithm.

The remainder of this paper is organized as follows. Section 2 presents the formulation details of MCDODE. Section 3 describes the solution pipeline and discusses practical issues for MCDODE framework. In section 4, a small network is used to demonstrate the accuracy, efficiency and robustness of the MCDODE framework. In addition, we demonstrate the scalability and computational efficiency of the proposed framework using real-world data in a large-scale network. At last, conclusions are drawn in section 5.

2. Formulation

In this section, we discuss the multi-class dynamic origin-destination estimation (MCDODE) framework. We first present the notations used in this paper, and then the multi-class dynamic network flow is modeled. Secondly, the MCDODE problem is formulated and represented on a computational graph. We then solve the MCDODE through a novel forward-backward algorithm.

2.1. Notations

Notations are summarized in Table 1.

Table 1: List of notations

A	The set of all links
K_q	The set of all OD pairs
K_{rs}	The set of all paths between OD pair rs
B	The set of indices of the observed flow
E	The set of indices of the observed travel time
H	The set of all time intervals
D	The set of vehicle classes
	Variables as scalars
h_1	The index of departure time interval of path flow or OD flow
h_2	The index of arrival time interval at the tail of link
i	index of vehicle class
$x_{ai}^{h_2}$	The flow arriving at the tail of link a in time interval h_2 for vehicle class i
$t_{ai}^{h_2}$	The link travel time of link a in time interval h_2 for vehicle class i
$q_{rsi}^{h_1}$	The flow of OD pair rs in time interval h_1 for vehicle class i
$f_{rsi}^{kh_1}$	The kth path flow for OD pair rs in time interval h_1 for vehicle class i
$p_{rsi}^{kh_1}$	The route choice portion of choosing path k in all paths between OD pair rs in time interval h_1 for vehicle class i

$c_{rsi}^{kh_1}$	The path travel time of kth path for OD pair rs in time interval h_1 for vehicle class i
$L_{ai}^{bh_2}$	The observation/link incidence for link a in time interval h_2 and observed flow b for vehicle class i
$M_{ai}^{eh_2}$	The weight of travel time for $t_{ai}^{h_2}$ in the observed travel time e
$\rho_{rsi}^{ka}(h_1, h_2)$	The portion of the kth path flow departing within time interval h_1 between OD pair rs which arrives at link a within time interval h_2 for vehicle class i (namely, an entry of the DAR matrix)
y_b	The bth observed flow, which might be a linear combination of link flow across vehicle classes, road segments and over time intervals
z_e	The eth observed travel time, which might be a linear combination of link travel time across vehicle classes, road segments and over time intervals
Variables as tensors	
$\mathbf{y}$	The vector of the observed flow
$\mathbf{z}$	The vector of the observed travel time
$\mathbf{x}_i$	The vector of link flow for vehicle class i
$\mathbf{t}_i$	The vector of link travel time for vehicle class i
$\mathbf{f}_i$	The vector of path flow for vehicle class i
$\mathbf{c}_i$	The vector of path travel time for vehicle class i
$\mathbf{L}_i$	The observation/link incidences matrix for vehicle class i
$\mathbf{M}_i$	The travel time weight matrix for vehicle class i
$\mathbf{p}_i$	The matrix of route choice portions for vehicle class i
ρ_i	The dynamic assignment ratio (DAR) matrix for vehicle class i

2.2. Modeling multi-class dynamic network flow

In this section, we first formulate a general network model for multi-class traffic flow in discrete time. We denote the path flow $f_{rsi}^{kh_1}$ as the number of class-i vehicles departing from kth path for OD pair rs in time interval h_1 and link flow $x_{ai}^{h_2}$ as the number of class-i vehicles arriving at the tail of link a in time interval h_2. The relationship between path flow and link flow is presented in Equation 1.

$$x_{ai}^{h_2} = \sum_{rs \in K_q} \sum_{k \in K_{rs}} \sum_{h_1 \in H} \rho_{rsi}^{ka}(h_1, h_2) f_{rsi}^{kh_1} \tag{1}$$

where K_q is the set of all OD pairs, and K_{rs} is the path set for OD pair rs. H is the set of all possible time intervals during study period. To avoid confusion, we denote departure time interval of path flow or OD flow as h_1, and the arrival time interval at the tail of link as h_2, respectively. The dynamic assignment ratio (DAR) $\rho_{rsi}^{ka}(h_1, h_2)$ denotes the portion of the kth path flow departing within time interval h_1 for OD pair rs which arrives at link a within time interval h_2 for vehicle class i (Ma & Qian 2018*b*). For each OD pair rs, there are K_{rs} paths for travelers to choose from, and the portion of choosing path k in time interval h_1 for vehicle class i is denoted as $p_{rsi}^{kh_1}$. The OD flow and path flow for vehicle class i can be represented by Equation 2.

$$f_{rsi}^{kh_1} = p_{rsi}^{kh_1} q_{rsi}^{h_1} \tag{2}$$

where OD demand $q_{rsi}^{h_1}$ represents the number of class-i vehicles for OD pair rs in time interval h_1. The link travel time, path travel time and DAR can be obtained from the dynamic network loading (DNL) models, as presented in Equation 3.

$$\left\{t_{ai}^{h_2}, c_{rsi}^{kh_1}, \rho_{rsi}^{ka}(h_1, h_2)\right\}_{r,s,i,k,a,h_1,h_2} = \Lambda(\{f_{rsi}^{kh_1}\}_{i,r,s,k,h_1}) \tag{3}$$

where $\Lambda(\cdot)$ represents the multi-class dynamic network loading models. The travel time can be generalized to any form of the disutility as long as it can be simulated by Λ. The generalized travel time can include roads tolls, left turn penalty, travelers' preferences and so on. The DNL function Λ takes the multi-class path flow as input and outputs the spatio-temporal network conditions $\{t_{ai}^{h_2}, c_{rsi}^{kh_1}\}$ and the DAR matrix $\rho_{rsi}^{ka}(h_1, h_2)$. Though there exists analytical solutions in small networks, Λ is usually represented by simulation-based models in large-scale networks. Many existing models including, but are not limited to, DynaMIT (Ben-Akiva et al. 1998), DYNASMART (Mahmassani et al. 1992), DTALite (Zhou & Taylor 2014) and MAC-POSTS (Ma et al. 2016, Pi et al. 2018), can be potentially used as function Λ.

The route choice portion $p_{rsi}^{kh_1}$ is obtained from a generalized route choice function, as presented in Equation 4.

$$p_{rsi}^{kh_1} = \Psi_{rsi}^{k}\left(\tilde{\mathbf{c}}_{rsi}^{kh_1}, \tilde{\mathbf{t}}_{ai}^{h_1}\right) \tag{4}$$

$$\tilde{\mathbf{c}}_{rsi}^{kh_1} = \left\{c_{rsi}^{kh'} | h' \leq h_1, h' \in H, rs \in K_q, k \in K_{rs}, i \in D\right\} \tag{5}$$

$$\tilde{\mathbf{t}}_{ai}^{h_1} = \left\{t_{ai}^{h'} | h' \leq h_1, h' \in H, a \in A, i \in D\right\} \tag{6}$$

where $\Psi_{rsi}^{kh_1}(\cdot,\cdot)$ is a generalized route choice model that takes in the path travel time and link travel time before time interval h_1 and computes the route choice portion for travelers in kth path of OD rs for vehicle class i. We further denote $t_{ai}^{h_2}$ as the travel time of link a in time interval h_2 for vehicle class i, and $c_{rsi}^{kh_1}$ as the travel time of path flow k in OD pair rs departing in time interval h_1. Most of the state-of-art route choice models (Prashker & Bekhor 2004, Zhou et al. 2012), including Logit and Probit models, satisfy Equation 4. The Logit and Probit models have been adopted in many studies and achieved great success in real-world applications (Maher et al. 2001). Combining Equation 1 and 2, we present the relation of link flow and OD flow in Equation 7.

$$x_{ai}^{h_2} = \sum_{rs \in K_q} \sum_{k \in K_{rs}} \sum_{h_1 \in H} \rho_{rsi}^{ka}(h_1, h_2) p_{rsi}^{kh_1} q_{rsi}^{h_1} \tag{7}$$

2.3. Modeling the observed flow and observed travel time

In this section, we describe the concept of the observed flow and observed travel time. The motivation for the definition of the observed flow and travel time lies in the indirect and aggregated observations of traffic networks. For example, loop detectors measure the traffic counts in each time interval, but the vehicle classes cannot be differentiated. The Department of Transportation regularly hire individuals to count the vehicles on road segments for both directions and different vehicle classes. In this case, the observation of traffic flow does not differentiate road directions.

To accommodate different kinds of flow observations, we propose the concept of observed flow, denoted by y_b, as a linear combination of link flow across vehicle classes, road segments and time intervals. The formulation of the observed flow is presented in Equation 8.

$$y_b = \sum_{i \in D} \sum_{a \in A} \sum_{h_2 \in H} L_{ai}^{bh_2} x_{ai}^{h_2} \tag{8}$$

where $b \in B$ is the index of observed flow and B is the set of indices of observed flow. The unit of y_b is the same as $x_{ai}^{h_2}$. For example, if there are two loop detectors recording the traffic counts in each interval, then we have $|B| = 2|H|$. By formulation 8, the observed flow can be the traffic count observations in various forms, including traffic counts of a single-class vehicles, aggregated traffic counts of all vehicles, aggregated traffic counts of a road for both directions, and aggregated traffic counts of multiple links. The link/observation incidence $L_{ai}^{bh_2}$ represents how the observed flow is aggregated. $L_{ai}^{bh_2}$ is 1 if link flow $x_{ai}^{h_2}$ is observed in the observed flow y_b and 0 otherwise, and a similar definition can be found in Yang et al. (2018).

Similarly, we may also observe the travel time across multiple links, vehicle classes and time intervals. For example, we may observe the total travel time of a highway which consists of multiple consecutive links, or we may observe the average travel time of car and trucks in a single link. To accommodate different kinds of travel time observations, we assume the observed travel time can be represented by a linear combination of all link travel time, as presented in Equation 9.

$$z_e = \sum_{i\in D}\sum_{a\in A}\sum_{h_2\in H} M_{ai}^{eh_2} t_{ai}^{h_2} \tag{9}$$

where $M_{ai}^{eh_2}$ represents the weight of travel time for $t_{ai}^{h_2}$ in the observed travel time e. The unit of z_e is the same as $t_{ai}^{h_2}$. $e \in E$ is the index of observed travel time and E is the set of indices of observed travel time. The formulation 9 is also general enough to accommodate various types of travel time observations. For example, the observed travel time includes link travel time of a single-class vehicles, path travel time of single-class vehicles, average travel time of all vehicle classes. We note that $M_{ai}^{eh_2} \in [0, 1]$ since we may observe the average travel time of multiple links, while $L_{ai}^{bh_2} \in \{0, 1\}$ because traffic flow is usually observed in an aggregated manner.

Example 1. *To illustrate the formulation of the observed flow and observed travel time, we consider a two-link network presented in Figure 1. We only consider one time interval, hence* $|H| = 1$. *We assume vehicle class* 1 *represents cars, and vehicle class* 2 *represents trucks.*

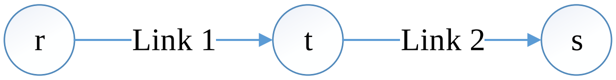

Figure 1: A two-link network

Suppose we observe 50 *cars passing link* 1 *and* 150 *vehicles (trucks and cars) passing link* 2*, then we have*

$$\begin{aligned} y_1 &= 50 \\ y_2 &= 150 \\ L_{a=1,i=1}^{b=1,h_2=1} &= 1 \\ L_{a=1,i=2}^{b=1,h_2=1} &= 0 \\ L_{a=2,i=1}^{b=2,h_2=1} &= 1 \\ L_{a=2,i=2}^{b=2,h_2=1} &= 1 \end{aligned}$$

Therefore

$$\begin{aligned} y_1 &= L_{a=1,i=1}^{b=1,h_2=1} x_{a=1,i=1}^{h_2=1} + L_{a=1,i=2}^{b=1,h_2=1} x_{a=1,i=2}^{h_2=1} \\ &= x_{a=1,i=1}^{h_2=1} \\ y_2 &= L_{a=2,i=1}^{b=2,h_2=1} x_{a=2,i=1}^{h_2=1} + L_{a=2,i=2}^{b=2,h_2=1} x_{a=2,i=2}^{h_2=1} \\ &= x_{a=2,i=1}^{h_2=1} + x_{a=2,i=2}^{h_2=1} \end{aligned}$$

Similarly, if we observe the travel time for traversing link 1 *and link* 2 *is* 100 *for cars, and the average travel time of link* 2 *is* 70 *for cars and trucks.*

$$\begin{aligned} z_1 &= 100 \\ z_2 &= 70 \\ M_{a=1,i=1}^{e=1,h_2=1} &= 1 \\ M_{a=2,i=1}^{e=1,h_2=1} &= 1 \\ M_{a=2,i=1}^{e=2,h_2=1} &= 0.5 \\ M_{a=2,i=2}^{e=2,h_2=1} &= 0.5 \end{aligned}$$

Then we have

$$\begin{aligned} z_1 &= M_{a=1,i=1}^{e=1,h_2=1} t_{a=1,i=1}^{h_2=1} + M_{a=2,i=1}^{e=1,h_2=1} t_{a=2,i=1}^{h_2=1} \\ &= t_{a=1,i=1}^{h_2=1} + t_{a=2,i=1}^{h_2=1} \\ z_2 &= M_{a=2,i=1}^{e=2,h_2=1} t_{a=2,i=1}^{h_2=1} + M_{a=2,i=2}^{e=2,h_2=1} t_{a=2,i=2}^{h_2=1} \\ &= \frac{t_{a=2,i=1}^{h_2=1} + t_{a=2,i=2}^{h_2=1}}{2} \end{aligned}$$

We note the "average travel time for cars and trucks" means the "average" of car travel time and truck travel time, respectively. It is also possible to consider the weighted average of travel time by car and truck flow together, which will be discussed in section 2.6.

2.4. MCDODE formulation

The formulation for multi-class dynamic origin-destination estimation (MCDODE) is presented in Equation 10.

$$\begin{aligned} \min_{\{q_{rsi}^{h_1}\}_{i,r,s,h_1}} \quad & w_1 \sum_{b\in B} \left(y'_b - \sum_{i\in D} \sum_{a\in A} \sum_{h_2\in H} L_{ai}^{bh_2} \left(\sum_{rs\in K_q} \sum_{k\in K_{rs}} \sum_{h_1\in H} \rho_{rsi}^{ka}(h_1,h_2) p_{rsi}^{kh_1} q_{rsi}^{h_1} \right)\right)^2 + w_2 \sum_{e\in E} \left(z'_e - \sum_{i\in D} \sum_{a\in A} \sum_{h_2\in H} M_{ai}^{eh_2} t_{ai}^{h_2} \right)^2 \\ \text{s.t.} \quad & \left(\{t_{ai}^{h_2}, c_{rsi}^{kh_1}, \rho_{rsi}^{ka}(h_1,h_2)\}\right)_{r,s,i,k,a,h_1,h_2} = \Lambda(\{f_{rsi}^{kh_1}\}_{i,r,s,k,h_1}) \\ & f_{rsi}^{kh_1} = p_{rsi}^{kh_1} q_{rsi}^{h_1}, \forall rs \in K_q, k \in K_{rs}, i \in D, h_1 \in H \\ & p_{rsi}^{kh_1} = \Psi_{rsi}^{k}\left(\tilde{\mathbf{c}}_{rsi}^{kh_1}, \tilde{\mathbf{t}}_{ai}^{h_1}\right), \forall rs \in K_q, k \in K_{rs}, i \in D, h_1 \in H \\ & q_{rsi}^{h_1} \geq 0, \forall rs \in K_q, h_1 \in H, i \in D \end{aligned} \tag{10}$$

where y'_b and z'_e are the observed flow and travel time data collected from real-world. The link/path travel time can be computed through tracking cumulative curves in the DNL model (Lu et al. 2013), and the computation for DAR will be discussed in section 3.2. The parameters w_1 and w_2 are the weight for each objective, respectively.

Equation 10 is a bi-level optimization problem with upper level minimizing the ℓ^2 norm between observed and estimated flow and travel time, and the lower level solves the traffic assignment problem denoted by function $\Psi_{rsi}^{k}(\cdot)$ and $\Lambda(\cdot)$. When $\Psi_{rsi}^{k}(\cdot,\cdot)$ represents the Dynamic User Equilibrium (DUE) conditions, Equation 10 is a Mathematical Program with Equilibrium Constraints (MPEC) problem. In contrast, when $\Psi_{rsi}^{k}(\cdot,\cdot)$ represents the Logit-model, Equation 10 can be formulated as either a bi-level optimization problem or a single-level non-linear optimization problem (Davis 1994, Ma & Qian 2018*a*).

Additional data sources, such as the historical OD demand and survey data, can also be used in the demand estimation. The DODE methods with these data have been extensively studied (Zhang et al. 2008, Wu et al. 2018). This paper focuses on the computational graph approach with observed traffic flow and travel time data, while other data can be potentially incorporated into the proposed framework.

The solution algorithm for the bi-level problem has been extensively studied over the past few decades. Various heuristic methods have been developed to solve the bi-level problem efficiently. For example, heuristic iterative algorithms between lower and upper problem are widely used in traffic applications (Yang 1995). Sensitivity analysis of the traffic assignment problem has been adopted to evaluate the gradients of bi-level problem. Lu et al. (2013) estimated the gradient of path flow using the dynamic network loading models, and Balakrishna et al. (2008) approximated the gradient of OD demand through stochastic perturbations. Osorio (2019) approximate the gradient by linearizing the dynamic traffic assignment models. All the studies mentioned above aim at finding the gradients of the OD demand for the bi-level formulation. In this paper, we propose to evaluate the gradient of OD demand analytically through the forward-backward algorithm on a computational graph, and details will be described in section 2.6.

2.5. Vectorizing the MCDODE formulation

First of all, the variables involved in the MCDODE formulation are vectorized, and the vectorization is performed for each vehicle class separately. We set $N = |H|$ and denote the total number path as $\Pi = \sum_{rs} |K_{rs}|$, $K = |K_q|$. The vectorized variables are presented in Table 2.

Table 2: MCDODE framework variable vectorization

Variable	Scalar	Vector	Dimension	Type	Description
OD flow	q^{h}_{rsi}	$\mathbf{q}_i$	$\mathbb{R}^{N\|K\|}$	Dense	q^{h}_{rsi} is place at entry $(h-1)\|K\|+k$
Path flow	f^{kh}_{rsi}	$\mathbf{f}_i$	$\mathbb{R}^{N\Pi}$	Dense	f^{kh}_{rsi} is placed at entry $(h-1)\Pi+k$
Link flow	x^{h}_{ai}	$\mathbf{x}_i$	$\mathbb{R}^{N\|A\|}$	Dense	x^{h}_{ai} is placed at entry $(N-1)\|A\|+k$
Link travel time	t^{h}_{ai}	$\mathbf{t}_i$	$\mathbb{R}^{N\|A\|}$	Dense	t^{h}_{ai} is placed at entry $(N-1)\|A\|+k$
Path travel time	c^{kh}_{rsi}	$\mathbf{c}_i$	$\mathbb{R}^{N\Pi}$	Dense	c^{kh}_{rsi} is placed at entry $(N-1)\Pi+k$
Observed flow	y_b	$\mathbf{y}$	$\mathbb{R}^{\|B\|}$	Dense	y_b is placed at entry b
Observed travel time	z_e	$\mathbf{z}$	$\mathbb{R}^{\|E\|}$	Dense	z_e is placed at entry e
DAR matrix	$\rho^{ka}_{rsi}(h_1,h_2)$	ρ_i	$\mathbb{R}^{N\|A\|\times N\Pi}$	Sparse	$\rho^{ka}_{rsi}(h_1,h_2)$ is placed at entry $[(h_2-1)\|A\|+a,(h_1-1)\Pi+k]$
Route choice matrix	p^{kh}_{rsi}	$\mathbf{p}_i$	$\mathbb{R}^{N\Pi\times N\|K\|}$	Sparse	p^{kh}_{rsi} is placed at entry $[(h-1)\|\Pi\|+k,(h-1)\|K\|+rs]$
Observation/link incidence matrix	L^{bh}_{ai}	$\mathbf{L}_i$	$\mathbb{R}^{\|B\|\times N\|\|A\|}$	Sparse	L^{bh}_{ai} is placed at entry $[b,(h-1)\|A\|+a]$
Link travel time portion matrix	M^{eh}_{ai}	$\mathbf{M}_i$	$\mathbb{R}^{\|E\|\times N\|\|A\|}$	Sparse	M^{eh}_{ai} is placed at entry $[e,(h-1)\|A\|+a]$

With the vectorized variables, Equations 7, 8 and 9 can be rewritten in Equation 11.

$$\begin{array}{rcl} \mathbf{x}_i &=& \rho_i\mathbf{p}_i\mathbf{q}_i, \forall i\in D \\ \mathbf{y} &=& \sum_{i\in D}\mathbf{L}_i\mathbf{x}_i \\ \mathbf{z} &=& \sum_{i\in D} M_i\mathbf{t}_i \end{array} \tag{11}$$

Multiplications between sparse matrix and sparse matrix, sparse matrix and dense vector are very efficient, especially on multi-core CPUs and Graphics Processing Units (GPUs). Therefore, Equation 11 can be evaluated efficiently. The MCDODE formulation in Equation 10 can be cast into the vectorized form presented in Equation 12.

$$\begin{array}{rrcll} \min\limits_{\{\mathbf{q}_i\}_i} & \multicolumn{4}{l}{w_1\left(\left\|\mathbf{y}'-\sum_{i\in D}\mathbf{L}_i\mathbf{x}_i\right\|_2^2\right)+w_2\left(\left\|\mathbf{z}'-\sum_{i\in D}\mathbf{M}_i\mathbf{t}_i\right\|_2^2\right)} \\ \text{s.t.} & \{\mathbf{t}_i,\mathbf{c}_i,\rho_i\}_i &=& \Lambda(\{\mathbf{f}_i\}_i) & \\ & \mathbf{f}_i &=& \mathbf{p}_i\mathbf{q}_i & \forall i\in D \\ & \mathbf{x}_i &=& \rho_i\mathbf{f}_i & \forall i\in D \\ & \mathbf{p}_i &=& \Psi_i(\{\mathbf{c}_i\}_i,\{\mathbf{t}_i\}_i) & \forall i\in D \\ & \mathbf{q}_i &\geq& 0 & \forall i\in D \end{array} \tag{12}$$

where $\Psi_i(\{\mathbf{c}_i\}_i,\{\mathbf{t}_i\}_i)$ is the vectorized route choice function for vehicle class i. We further substitute the path flow and link flow to the objective function, as presented in Equation 13.

$$\begin{array}{rrcll} \min\limits_{\{\mathbf{q}_i\}_i} & \multicolumn{4}{l}{w_1\left(\left\|\mathbf{y}'-\sum_{i\in D}\mathbf{L}_i\rho_i\mathbf{p}_i\mathbf{q}_i\right\|_2^2\right)+w_2\left(\left\|\mathbf{z}'-\sum_{i\in D}\mathbf{M}_i\mathbf{t}_i\right\|_2^2\right)} \\ \text{s.t.} & \{\mathbf{t}_i,\mathbf{c}_i,\rho_i\}_i &=& \Lambda(\{\mathbf{f}_i\}_i) & \\ & \mathbf{p}_i &=& \Psi_i(\{\mathbf{c}_i\}_i,\{\mathbf{t}_i\}_i) & \forall i\in D \\ & \mathbf{q}_i &\geq& 0 & \forall i\in D \end{array} \tag{13}$$

2.6. A computational graph for MCDODE

In order to solve the MCDODE problem, our goal is to obtain the gradient of OD demand for formulation 13. In this section, we propose a novel approach to obtain the gradient of OD demand through the forward-backward algorithm on a computational graph. First we cast equation 13 into a computational graph representation, and Figure 2 describes the structure of the computational graph for MCDODE. The forward-backward algorithm runs on the computational graph, and algorithm consists of two processes: the forward iteration and the backward iteration.

In general, the forward iteration assumes that OD demand is fixed and it solves for the network conditions, while the backward iteration assumes the network conditions are fixed and it updates the OD demand. The whole process of forward-backward algorithm resembles some heuristic methods that solve the upper level and lower level problem iteratively (Yang 1995). However, the proposed algorithm can evaluate the instantaneous gradient of OD demand analytically in the backward iteration.

Forward iteration: Forward iteration takes multi-class OD demand as the input and solves the traffic assignment problem presented in Equation 14.

$$\begin{array}{lcll}
\mathbf{f}_i & = & \mathbf{p}_i\mathbf{q}_i & \forall i \in D \\
\{\mathbf{t}_i, \mathbf{c}_i, \rho_i\}_i & = & \Lambda(\{\mathbf{f}_i\}_i) & \\
\mathbf{p}_i & = & \Psi_i(\{\mathbf{c}_i\}_i, \{\mathbf{t}_i\}_i) & \forall i \in D
\end{array} \tag{14}$$

The forward iteration includes the multi-class dynamic network loading models Λ and route choice models $\{\Psi_i\}_i$. The output of the forward iteration is route choice portion $\mathbf{p}_i$, DAR matrix ρ_i and link/path travel time $(\mathbf{t}_i, \mathbf{c}_i)$. We omit the solution method for traffic assignment problem, as it has been extensively studied in many literature (Peeta & Ziliaskopoulos 2001). After solving the dynamic traffic assignment models, the forward iteration compute the objective function (loss) in formulation 13, which can be represented by a series of equations in Equation 15 and 16.

$$\mathcal{L} = w_1\mathcal{L}_1 + w_2\mathcal{L}_2 \tag{15}$$

$$\begin{array}{lcl@{\qquad}lcl}
\mathcal{L}_1 & = & \|\mathbf{y}' - \mathbf{y}\|_2^2 & \mathcal{L}_2 & = & \|\mathbf{z}' - \mathbf{z}\|_2^2 \\
\mathbf{y} & = & \sum_{i\in D}\mathbf{L}_i\mathbf{x}_i & \mathbf{z} & = & \sum_{i\in D}\mathbf{M}_i\mathbf{t}_i \\
\mathbf{x}_i & = & \rho_i\mathbf{f}_i & \{\mathbf{t}_i\}_i & = & \bar{\Lambda}(\{\mathbf{x}_i\}_i) \\
\mathbf{f}_i & = & \mathbf{p}_i\mathbf{q}_i & \mathbf{x}_i & = & \rho_i\mathbf{f}_i \\
 & & & \mathbf{f}_i & = & \mathbf{p}_i\mathbf{q}_i
\end{array} \tag{16}$$

where $\mathbf{y}'$ is the observed flow and $\mathbf{y}$ is the reproduction of the observed flow estimated from the traffic assignment model. Similarly, $\mathbf{z}'$ is the observed travel time and $\mathbf{z}$ is the reproduction of the observed travel time estimated from the traffic assignment model. We use $\bar{\Lambda}$ to represent a part of the function Λ, and $\bar{\Lambda}$ takes dynamic link flow as input and outputs the link travel time $\{\mathbf{t}_i\}_i$. Precisely, $\bar{\Lambda}$ represents the dynamic link models (Zhang et al. 2013, Jin 2012).

Backward iteration: The backward iteration searches for the gradient of OD demand for formulation 17 with the route choice portion $\mathbf{p}_i$, DAR matrix ρ_i and travel time $\{\mathbf{c}_i, \mathbf{t}_i\}_i$ known from the forward iteration.

$$\begin{array}{ll}
\min\limits_{\{\mathbf{q}_i\}_i} & w_1\left(\left\|\mathbf{y}' - \sum\limits_{i\in D}\mathbf{L}_i\rho_i\mathbf{p}_i\mathbf{q}_i\right\|_2^2\right) + w_2\left(\left\|\mathbf{z}' - \sum\limits_{i\in D}\mathbf{M}_i\mathbf{t}_i\right\|_2^2\right) \\
\mathbf{q}_i \geq 0 & \forall i \in D
\end{array} \tag{17}$$

When the gradient of OD demand is known, a projected gradient descent method can be used to solve Equation 17. The reason we call the solution process for Equation 17 a "backward iteration" is that, the gradient of OD demand for Equation 17 can be evaluated through the backpropagation (BP) method. Taking the derivative of the objective function step by step, we have a series of equations presented in Equation 18

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \mathcal{L}_1} &= w_1 & \frac{\partial \mathcal{L}}{\partial \mathcal{L}_2} &= w_2 \\
\frac{\partial \mathcal{L}_1}{\partial \mathbf{y}} &= 2\left(\mathbf{y}' - \textstyle\sum_{i' \in D} \mathbf{L}_{i'} \rho_{i'} \mathbf{p}_{i'} \mathbf{q}_{i'}\right) & \frac{\partial \mathcal{L}_2}{\partial \mathbf{z}} &= 2\left(\mathbf{z}' - \textstyle\sum_{i' \in D} \mathbf{M}_{i'} \mathbf{t}_{i'}\right) \\
\frac{\partial \mathcal{L}_1}{\partial \mathbf{x}_i} &= -\mathbf{L}_i^T \frac{\partial \mathcal{L}_1}{\partial \mathbf{y}} & \frac{\partial \mathcal{L}_2}{\partial \mathbf{t}_i} &= -\mathbf{M}_i^T \frac{\partial \mathcal{L}_2}{\partial \mathbf{z}} \\
\frac{\partial \mathcal{L}_1}{\partial \mathbf{f}_i} &= \rho_i^T \frac{\partial \mathcal{L}_1}{\partial \mathbf{x}_i} & \frac{\partial \mathcal{L}_2}{\partial \mathbf{x}_i} &= \frac{\partial \bar{\Lambda}(\{\mathbf{x}_i\}_i)}{\partial \mathbf{x}_i} \frac{\partial \mathcal{L}_2}{\partial \mathbf{t}_i} \\
\frac{\partial \mathcal{L}_1}{\partial \mathbf{q}_i} &= \mathbf{p}_i^T \frac{\partial \mathcal{L}_1}{\partial \mathbf{f}_i} & \frac{\partial \mathcal{L}_2}{\partial \mathbf{f}_i} &= \rho_i^T \frac{\partial \mathcal{L}_2}{\partial \mathbf{x}_i} \\
& & \frac{\partial \mathcal{L}_2}{\partial \mathbf{q}_i} &= \mathbf{p}_i^T \frac{\partial \mathcal{L}_2}{\partial \mathbf{f}_i}
\end{aligned}
\tag{18}
$$

Combining the equations in 18, the gradient of OD demand is presented in Equation 19.

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \mathbf{q}_i} &= \frac{\partial \mathcal{L}}{\partial \mathcal{L}_1} \frac{\partial \mathcal{L}_1}{\partial \mathbf{q}_i} + \frac{\partial \mathcal{L}}{\partial \mathcal{L}_2} \frac{\partial \mathcal{L}_2}{\partial \mathbf{q}_i} \\
&= -2 w_1 \mathbf{p}_i^T \rho_i^T \mathbf{L}_i^T \left(\mathbf{y}' - \textstyle\sum_{i' \in D} \mathbf{L}_{i'} \rho_{i'} \mathbf{p}_{i'} \mathbf{q}_{i'}\right) - 2 w_2 \mathbf{p}_i^T \rho_i^T \frac{\partial \bar{\Lambda}(\{\mathbf{x}_i\}_i)}{\partial \mathbf{x}_i} \mathbf{M}_i^T \left(\mathbf{z}' - \textstyle\sum_{i' \in D} \mathbf{M}_{i'} \mathbf{t}_{\prime}\right)
\end{aligned}
\tag{19}
$$

The forward-backward algorithm is illustrated in Figure 2. The solid arrow represents the forward iteration and the dashed arrow represents the backward iteration. We omit the forward and backward iterations for historical OD data, while the other processes are described above. The forward and backward iterations for historical OD data are straightforward by adding $w_3 \mathcal{L}_3 = w_3 \sum_{i \in D} \left\| \mathbf{q}_i' - \mathbf{q}_i \right\|_2^2$ to the MCDODE formulation 13 (Zhang et al. 2008).

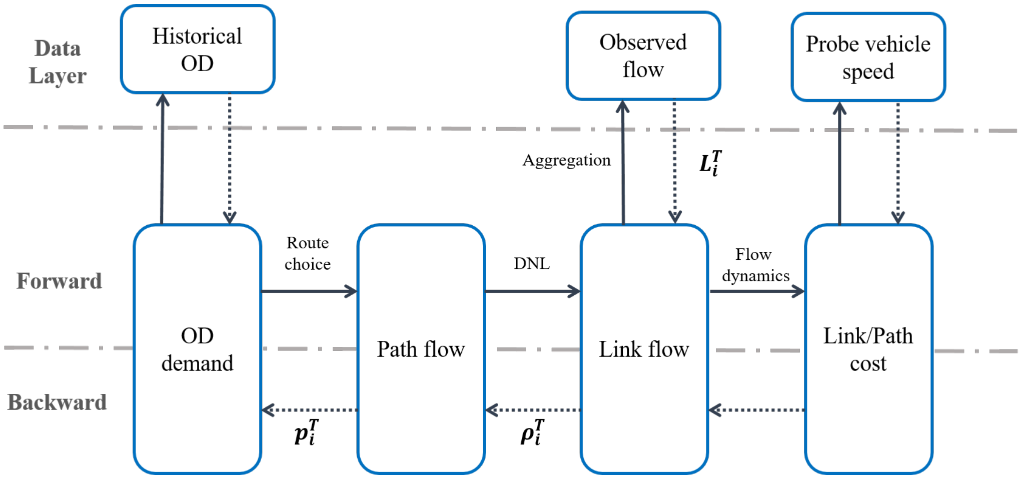

Figure 2: An illustration of the forward-backward algorithm.

To solve the MCDODE formulation in Equation 12, we run the forward-backward algorithm to compute the gradient of OD demand. Then the projected gradient descent method is used to update the OD demand. There is a family of gradient-based methods that can be used, and we will discuss them in section 3.4. Comparisons among different methods will also be conducted in the numerical experiments.

Our computational graph approach shares many similarities with the deep learning models: 1) both models contain high dimensional parameters; 2) multi-core CPU and GPU can be used to speed up the solution process; 3) many advanced variants of gradient-based method can be used to solve the models; 4) Backpropagation method can be used to evaluate the gradient layer by layer (Rumelhart et al. 1985, Wu et al. 2018). Potentially, all the techniques used in the training for deep learning can be used for the proposed computational graph. In this paper, we will test the advanced gradient-based method and multi-processing, while leaving other techniques, such as dropout (Gal & Ghahramani 2016), transfer learning (Pan & Yang 2010) and regularization (Neyshabur et al. 2014) for future research.

There are many flexible ways to incorporate observed data in the backward-forward algorithm. For example, it is possible to view the vehicle trajectories as samples from path flow and incorporate the trajectory data into the computational graph. We can also compute the weighted average speed for cars and trucks to reproduce the "average link travel time" since the network conditions (flow, travel time) are fixed in the backward iterations. Hence the question left in Example 1 is answered.

As for the stopping criteria, we first claim Proposition 1 holds by the definition of the forward-backward algorithm.

Proposition 1. *In the proposed forward-backward algorithm, the DAR matrix, route choice portions and link/path travel time do not change if and only if* $\frac{\partial \mathcal{L}}{\partial \mathbf{q}_i} = 0, \forall i$ *during the forward and backward iterations.*

Proposition 1 indicates that the forward-backward algorithm converges when either of the following two conditions hold: 1) in the forward iteration, the DAR matrix, route choice portions and link/path travel time do not change; 2) in the backward iteration, $\frac{\partial \mathcal{L}}{\partial \mathbf{q}_i} = 0$. During the forward-backward algorithm, we can either monitor the change of DAR matrix, route choice portions and link/path or the change of the gradient of OD demand. We can also see that the forward-backward algorithm converges to the local minimum for formulation 13 in Proposition 2.

Proposition 2. *When the forward-backward algorithm converges, the estimated OD demand* $\{\mathbf{q}_i\}_i$ *is a local minimum for formulation 13.*

Proof. When the forward-backward algorithm converges, we know $\frac{\partial \mathcal{L}}{\partial \mathbf{q}_i} = 0$ by Proposition 1. Since formulation 17 is convex, $\frac{\partial^2 \mathcal{L}}{\partial \mathbf{q}_i^2} \geq 0$. Therefore, $\{\mathbf{q}_i\}_i$ is the local minimum solution. □

3. Solution Algorithms

In this section, we first discuss several practical issues to complete the MCDODE framework with the forward-backward algorithm. We develop a multi-class traffic simulation package called MAC-POSTS for the network loading function $\Lambda(\cdot)$, and a Growing Tree algorithm is proposed to obtain the DAR matrix. Secondly, we discuss how to evaluate the derivative of link travel time in dynamic networks. Thirdly, we present how to incorporate multi-day observation data in the proposed MCDODE framework with multiprocessing. Lastly, the whole framework for MCDODE is presented.

3.1. Multi-class traffic simulation

The dynamic network loading function $\Lambda(\cdot)$ is fulfilled with the mesoscopic multi-class traffic simulation package Mobility Data Analytics Center - Prediction, Optimization, and Simulation toolkit for Transportation Systems (MAC-POSTS) developed by the Mobility Data Analytics Center at Carnegie Mellon University. To simulate heterogeneous traffic with multi-class vehicles like cars and trucks, the simulation package captures flow dynamics and outputs the traffic metrics for multi-class vehicles. The flow metrics include the traffic volumes, traffic speed and travel time. Due to the page limitation, more details about the mesoscopic multi-class traffic simulation model in MAC-POSTS is presented in (Qian et al. 2017, Pi et al. 2018).

With the multi-class dynamic OD demand known, the mesoscopic multi-class traffic simulation model performs the following steps orderly in every loading interval (*e.g.* five seconds):

1. Vehicle generation: multi-class vehicles are generated at origins according to the traffic demand.
2. Routing: the route choice behaviors of all vehicles are updated, according to the route choice models.
3. Node evolution: cars and trucks are moved through intersections following the intersection flow model.
4. Link evolution: cars and trucks are moved on links following the link flow model.
5. Network flow statistics: the model records link flow counts, link speeds, travel time and other network performance statistics.

We note the loading interval is different from the interval defined in this paper, since loading interval is usually much shorter. After running the simulation, the route choice portion $\mathbf{p}_i$ and link/route travel time $(\mathbf{t}_i, \mathbf{c}_i)$ for each vehicle class can be obtained based on the simulation results. The DAR matrix ρ_i for each vehicle class can also be obtained by constructing the tree-based cumulative curves during the simulation process, and details are presented in section 3.2.

3.2. Tree-based cumulative curve

In this section, we develop a novel method to compute the DAR matrix during the traffic simulation. Computing the DAR matrix during the simulation is more efficient than obtaining the DAR matrix after the simulation. However, computing the dynamic assignment ratio (DAR) during the traffic simulation is challenging. A naive method to obtain the DAR matrix is by recording the trajectories of all simulated vehicles in the DNL process. This method iterates across all paths and links over all time intervals, and in each iteration the method computes the number of vehicles arriving at a specific link from a specific path. Since the dimension of DAR matrix increases exponentially with respect to the size of network and the number of time intervals, the naive method is computational implausible for large-scale networks. In this section, we propose a novel method to compute the DAR matrix through the tree-based cumulative curves, and the proposed method is efficient in both computational time (time complexity) and memory (space complexity).

We define $\chi_{arsi}^{kh_1}(\cdot)$ as the tree-based cumulative curve of link a for class-i vehicles departing from path k in OD pair rs in time interval h_1. $\chi_{arsi}^{kh_1}(t)$ takes the time t as input and outputs the total number of vehicles departing in time interval h_1 from path rsk and arriving at link a before time t. Then the DAR can be computed by Equation 20.

$$\rho_{rsi}^{ka}(h_1, h_2) = \frac{\chi_{arsi}^{kh_1}(\overline{t_2}) - \chi_{arsi}^{kh_1}(\underline{t_2})}{f_{rsi}^{kh_1}} \tag{20}$$

where $\underline{t_2}$ is the beginning of time interval h_2 and $\overline{t_2}$ is the end of time interval h_2.

We note that $\chi_{arsi}^{kh_1}(\cdot)$ records the cumulative curves for each path flow and departing time interval separately, and hence it requires more memory and computational power than the standard link-based cumulative curve (Lu et al. 2013). However, only a very small fraction of vehicles pass a specific link a during the simulation. There are only a small fraction of paths containing a specific link, and hence the $\chi_{arsi}^{kh_1}(\cdot)$ is sparse in terms of path indices rsk and time indices h_1. Using this intuition, we develop a Growing Tree algorithm to build the tree-based cumulative curve $\chi_{arsi}^{kh_1}(\cdot)$ for each link a. Since the algorithm is tree-based, so we refer $\chi_{arsi}^{kh_1}(\cdot)$ as the tree-based cumulative curves.

The process of the Growing Tree algorithm is presented in Algorithm 1.

Algorithm 1: Growing Tree algorithm for constructing $\chi^{kh_1}_{arsi}(\cdot)$

```
GrowingTree ($S, n$);
Input  : Traffic simulator $S$, number of loading intervals $n$
Output: Tree-based cumulative curves $\chi$
Initialize an empty dictionary $\chi$;
for ($i = 0;\ i < n;\ ++i$) do
    Run the simulator $S$ for one loading interval;
    for $a \in A$ do
        Initialize an empty dictionary $\chi[a]$;
        Extract the set of vehicles going in link $a$ and denote it as $Q$;
        for $v \in Q$ do
            Suppose vehicle $v$ follows path $k$ in OD pair $rs$ and departs in time interval $h_1$ and the vehicle
              class is $i$;
            if $i$ is not the key of dictionary $\chi[a]$ then
                Initialize $\chi[a][i]$ with an empty dictionary;
            end
            if $h_1$ is not the key of dictionary $\chi[a][i]$ then
                Initialize $\chi[a][i][h_1]$ with an empty dictionary;
            end
            if $rsk$ is not the key of dictionary $\chi[a][i][h_1]$ then
                Initialize $\chi[a][i][h_1][rsk]$ with an empty cumulative curve;
            end
            Add record $(i, 1)$ to the cumulative curve $\chi[a][i][h_1][rsk]$;
        end
    end
end
```

In the algorithm, a record $(i, 1)$ means one vehicle arriving at the link a at time t. $\chi^{kh_1}_{arsi}(\cdot)$ is constructed as a tree, as presented in Figure 3. During the construction, when a vehicle transverses a link, a leaf containing a standard cumulative curve is either created or updated to record the location of that vehicle. The tree $\chi^{kh_1}_{arsi}(\cdot)$ is unbalanced, so a hashmap-based tree is more memory efficient. In Algorithm 1, a dictionary refers to a key-value mapping implemented by hashmap, and readers can view the dictionary as one of the following data structures: `dictionary` in Python, `HashMap` in Java, or `unordered_map` in C++.

3.3. Derivatives of link travel time

In the backward iteration described in section 2.6, we need to compute the derivatives of link travel time $\frac{\partial \bar{\Lambda}(\{\mathbf{X}_i\}_i)}{\partial \mathbf{X}_i}$. The function $\bar{\Lambda}$ represents the link flow models which include, but is not limited to, point queue, spatial queue, cell transmission model, link transmission model and link queue model (Jin 2012, Zhang et al. 2013). There is no closed-form for link flow models, hence the derivatives of function $\bar{\Lambda}$ can be challenging to evaluate analytically. In contrast, there exists methods to approximate the link travel time derivatives for the simulation-based models, and the basic idea of this kind of methods is to examine the extra link travel time induced by a marginal vehicle added to the link. Readers are refereed to Qian et al. (2012), Lu et al. (2013) for the implementation details. In this paper, we adopt the approximation approach discussed in Lu et al. (2013) to evaluate $\frac{\partial \bar{\Lambda}(\{\mathbf{X}_i\}_i)}{\partial \mathbf{X}_i}$.

3.4. Incorporating multi-day observations with multiprocessing

From formulation 13, the multi-class demand is estimated with one data sample $(\mathbf{y}', \mathbf{z}')$. In real world applications, we may observe the flow and travel time on multiple days. Suppose we collect data samples for M days and we let $(\mathbf{y}'_m, \mathbf{z}'_m)$ denote the observed flow and travel time on day m, then the MCDODE problem 13 can be extended

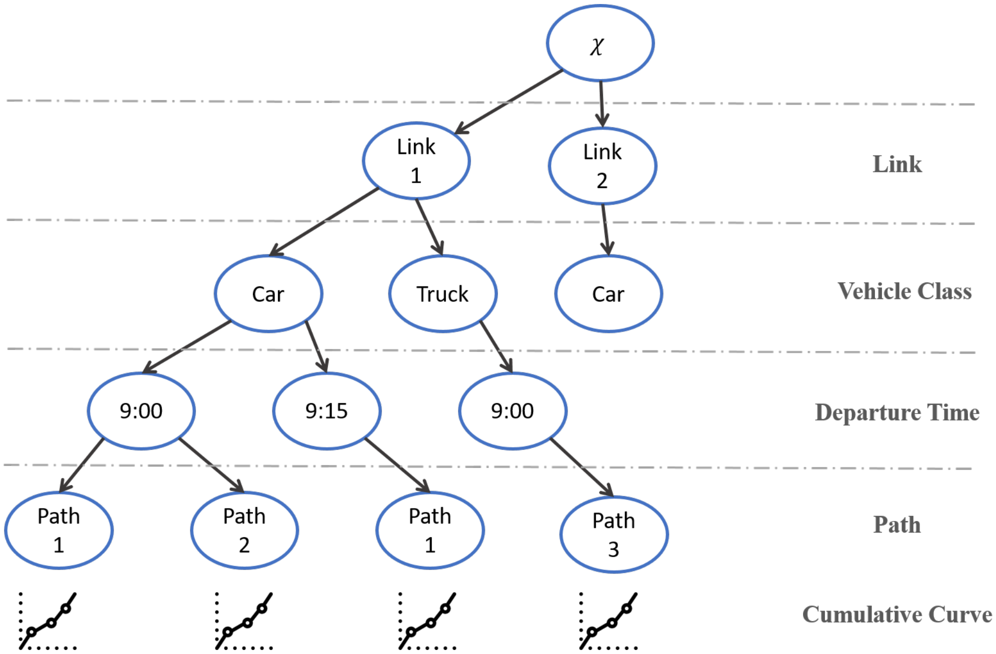

Figure 3: An illustration of the Growing Tree algorithm.

to accommodate multi-day observations in formulation 21

$$
\begin{array}{llll}
\min\limits_{\{\mathbf{q}_i\}_i} & \multicolumn{3}{l}{\frac{1}{M}\sum\limits_{1\le m\le M}\left[w_1\left(\left\|\mathbf{y}'_m-\sum\limits_{i\in D}\mathbf{L}_i\rho_i\mathbf{p}_i\mathbf{q}_i\right\|_2^2\right)+w_2\left(\left\|\mathbf{z}'_m-\sum\limits_{i\in D}\mathbf{M}_i\mathbf{t}_i\right\|_2^2\right)\right]} \\
\text{s.t.} & \{\mathbf{t}_i,\mathbf{c}_i,\rho_i\}_i = & \Lambda(\{\mathbf{f}_i\}_i) & \\
 & \mathbf{p}_i = & \Psi_i(\{\mathbf{c}_i\}_i,\{\mathbf{t}_i\}_i) & \forall i\in D \\
 & \mathbf{q}_i \geq & 0 & \forall i\in D
\end{array}
\tag{21}
$$

Formulation 21 can be directly solved using the forward-backward algorithm. We only need to compute the gradients of OD demand for each data sample and use the average gradient over all data samples to update the OD demand during the backward iteration. This process is the same as Gradient Descent (GD) method. In addition, the stochastic gradient descent (SGD) method can also be used to solve formulation 21. In the process of SGD, we evaluate the gradient of OD demand for one randomly selected data sample and then use it to update the OD demand. The comparisons between GD and SGD exist in many machine learning models, readers are refereed to (Saad 1998) for more details. Many advanced gradient descent methods can also be used to solve formulation 21. For example, Adagrad is one of the most representatives of variants of SGD with adaptive step sizes, and it is often used in the optimization of deep neural networks (Duchi et al. 2011).

We can further speed up the solution process for formulation 21 by utilizing the power of multiprocessing. The delayed stochastic gradient descent (delayed-SGD) method can evaluate the gradient of multiple data samples on a multi-core CPU at the same time (Zinkevich et al. 2009). Each core is responsible for evaluating one single data sample at one time. Comparing to the traditional SGD, the delayed-SGD makes full use of the multi-core CPU and hence it can solve the MCDODE framework more efficiently. It is also possible to extend Formulation 21 to incorporate multi-day data that are observed on different links. To do that, we can replace $\mathbf{L}_i$ and $\mathbf{M}_i$ to $\mathbf{L}_{im}$ and $\mathbf{M}_{im}$ for each day m separately.

3.5. Solution framework

The solution algorithm for MCDODE is summarized in Table 3.

Algorithm	**[*MCDODE-FRAMEWORK*]**
Step 0	*Initialization.* Initialize the OD demand vector $\{\mathbf{q}_i\}_i$ for each vehicle class.
Step 1	*Forward iteration.* Solve the traffic assignment model presented in equation 14 with OD demand $\{\mathbf{q}_i\}_i$, and construct the tree-based cumulative curve χ through Growing Tree algorithm presented in Algorithm 1.
Step 2	*Variable retrieval.* Extract the link/path travel time from the simulation model, compute the route choice matrix from route choice model by Equation 4, and obtain the DAR matrix from the tree-base cumulative curves by Equation 20.
Step 3	*Backward iteration.* Compute the gradient of OD demand using the backward iteration presented in Equation 18 and 19.
Step 4	*Update OD demand.* Update the OD demand with the gradient-based projection method discussed in section 3.4.
Step 5	*Convergence check.* Stop when the change of OD demand $\{\mathbf{q}_i\}_i$ is less than tolerance. Otherwise, go to Step 1.

Table 3: MCDODE solution framework

4. Numerical Experiments

In this section, we first examine the proposed MCDODE framework in a small network. Estimation results are presented and discussed. We examine the effects of multiprocessing, compare different variants of gradient descent methods, and conduct the sensitivity analysis of the MCDODE framework. In addition, the effectiveness, efficiency and scalability of the MCDODE framework are demonstrated in a large-scale network. All the experiments in this section are conducted on a desktop with Intel Core i7-6700K CPU 4.00GHz × 8, 2133 MHz 2 × 16GB RAM, 500GB SSD.

4.1. A small network

We first work with a small network with seven links, three paths and one O-D pair, as presented in Figure 4. Two classes of vehicles are considered: cars and trucks. Link 1 and Link 7 are OD connectors, and we use the identical triangular fundamental diagram (FD) for the rest of 5 links. In the FD, length of each road segment is 0.55 mile, free flow speed is 35 miles/hour for car and 25 miles/hour for truck, flow capacity is 2,200 vehicles/hour for car and 1,200 vehicles/hour for truck, and the holding capacity is 200 vehicles/mile for car and 80 vehicles/mile for truck.

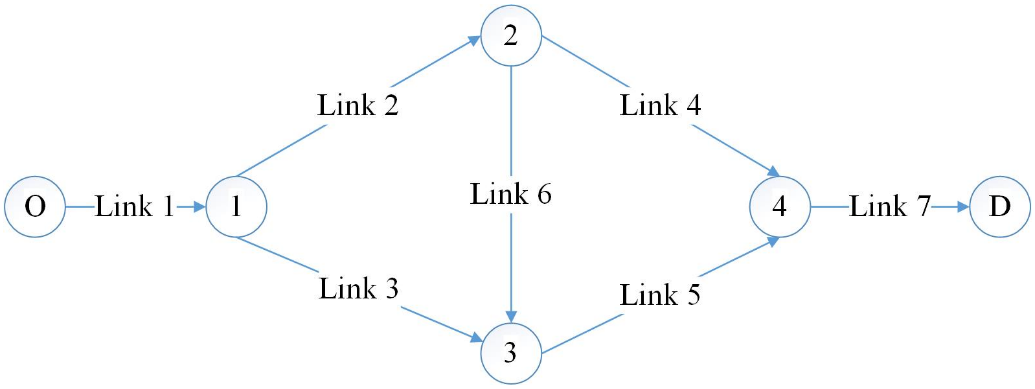

Figure 4: A small network.

We generate the multi-class dynamic OD demand by random number generators and then treat it as the "true" OD demand. We generate the observed flow by running the multi-class network simulation model and then adding the noise. The performance of the MCDODE estimation formulation is assessed by comparing the estimated flow with the "true" flow (flow includes observed flow, link flow, OD demand) (Antoniou, Barceló, Breen, Bullejos, Casas, Cipriani, Ciuffo, Djukic, Hoogendoorn, Marzano et al. 2015). We use R-square between the "true" flow (travel time) and estimated flow (travel time) to measure the estimation accuracy.

Baseline setting: in the small network, we randomly sample the "true" car and truck OD demand from uniform distributions $\texttt{Unif}(0, 300)$ and $\texttt{Unif}(0, 60)$, respectively. We randomly generate the route choice portions and treat them as unknown, then we run the MAC-POSTS Λ to obtain the "true" network conditions. We construct the observed flow as follows: firstly we randomly generate the matrix $\{\mathbf{L}_i\}_i$ by a Bernoulli distribution with $p = 0.5$ for link $3, 4, 5, 6$ and leave the flow of link 2 hidden from all observations; secondly we aggregate the "true" link flow to obtain the observed flow by $\{\mathbf{L}_i\}_i$; thirdly we multiply $1 + \varepsilon$ to the "true" observed flow to get the observed flow with noise, where $\varepsilon \sim \texttt{Unif}(-\xi, \xi)$ and $\xi \in [0, 1)$ represents the noise level. We consider 10 time intervals, and each time interval represents fifteen minutes. We set $|B| = 10$, and 6 observations are from car flow and 4 observations are from truck flow. Assuming we directly observe the travel time for link $3, 4, 5, 6$ for cars and trucks separately, $\{\mathbf{M}_i\}_i$ and E can be constructed accordingly. We set $w_1 = 1, w_2 = 0.01$. We also add noise to the observed link travel time using the same method as observed flow. We observe 8 data samples $M = 8$ and the noise level $\xi = 0.1$. We use single-process Adagrad with step size 1 as the solution method, and the initial OD demand is generated from $\texttt{Unif}(0, 15)$ and $\texttt{Unif}(0, 3)$ for car and trucks, respectively. The above setting is called the baseline setting.

4.1.1. Basic estimation results

In this section, we examine the basic estimation results of the proposed MCDODE framework for the baseline setting. We run the MCDODE framework presented in section 3.5 until convergence. The change of loss $\mathcal{L}$ against the number of iterations using Adagrad is presented in Figure 5.

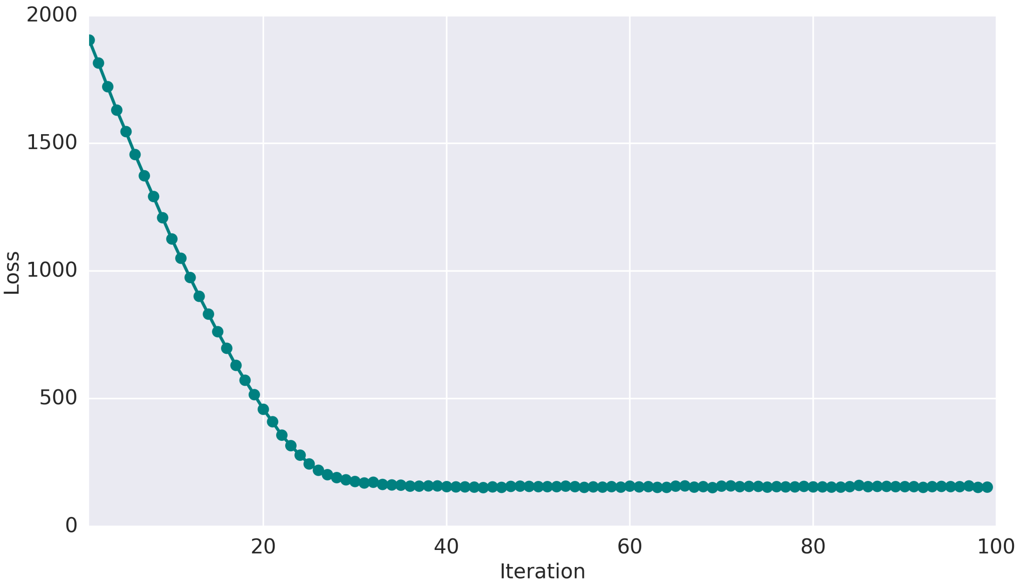

Figure 5: Convergence curve for the loss $\mathcal{L}$

To analyze the convergence of the observed flow and travel time separately, we decompose the loss $\mathcal{L}$ into four components: car flow, car travel time, truck flow and truck travel time. We plot the loss for the four components separately, and the results are presented in Figure 6. Note we normalize the loss such that it is between 0 and 1.

The comparisons between the "true" and estimated values for observed flow, link flow, link travel time and OD demand are presented in Figure 7, 8, 9 and 10, respectively.

The R-squares between the "true" and estimated flow (travel time) are presented in Table 4. The proposed MCDODE framework yields accurate estimation of the multi-class dynamic OD demand in the small network. The average R-squares for car flow and truck flow are above 0.98. The estimation accuracy for truck is lower than car, which is probably attributed to the low truck demand. Since the multi-class traffic loading model is discretized

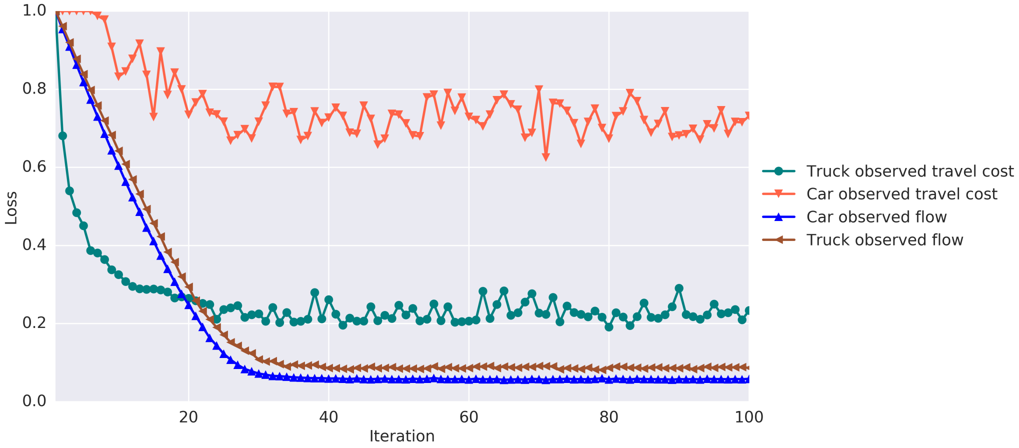

Figure 6: Decomposed convergence curve for observed flow and travel time (normalized)

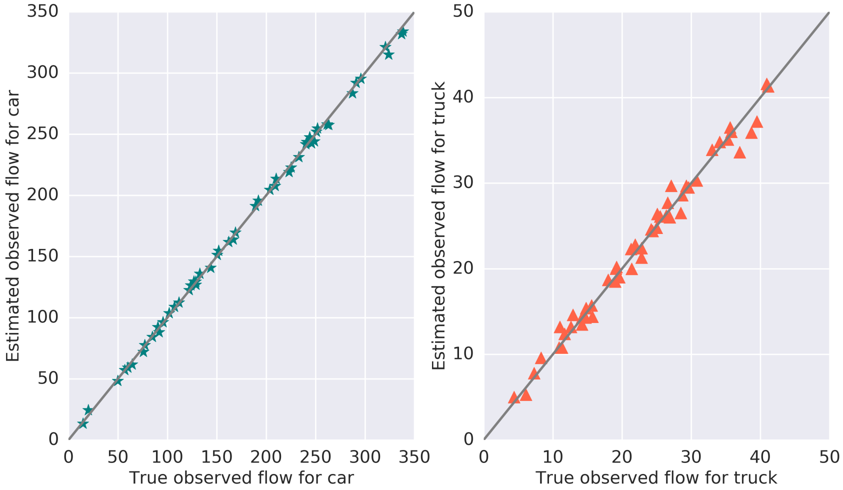

Figure 7: Estimated and "true" observed flow for cars and trucks (unit:vehicle/15mins).

and stochastic (Qian et al. 2017), low demand may incur a large variance in the simulation results. Therefore, the gradient of truck flow becomes noisy when the demand is low.

The R-square for the travel time is lower than that for flows, since the derivative of travel time $\frac{\partial \bar{\Lambda}(\{\mathbf{X}_i\}_i)}{\partial \mathbf{X}_i}$ is approximated by the simulation rather than evaluated in a closed-form. Again, due to the discretization and stochasticity of the simulation model, the approximations of $\frac{\partial \bar{\Lambda}(\{\mathbf{X}_i\}_i)}{\partial \mathbf{X}_i}$ can be noisy.

4.1.2. Comparing different gradient-based methods

In this section, we examine the performance of three gradient-based methods: gradient descent (GD), stochastic gradient descent (SGD) method and Adagrad. We solve the MCDODE problem for the baseline setting three times using different gradient-based methods, and we plot the convergence curves for the three methods in Figure 11.

For all three method, it takes less than 2 minutes to complete the 100 iterations. One can clearly see that SGD outperforms the GD throughout the 100 iterations. Though GD and SGD converge faster in the first 20 iterations, the Adagrad outperforms both methods in terms of convergence rate and final loss after 100 iterations. The reason for the best performance of Adagrad is probably because Adagrad can select the step size adaptively during the solution, hence it maintains a good convergence rate throughout the 100 iterations. Adagrad is adopted as the standard method to solve the MCDODE formulation in the baseline setting.

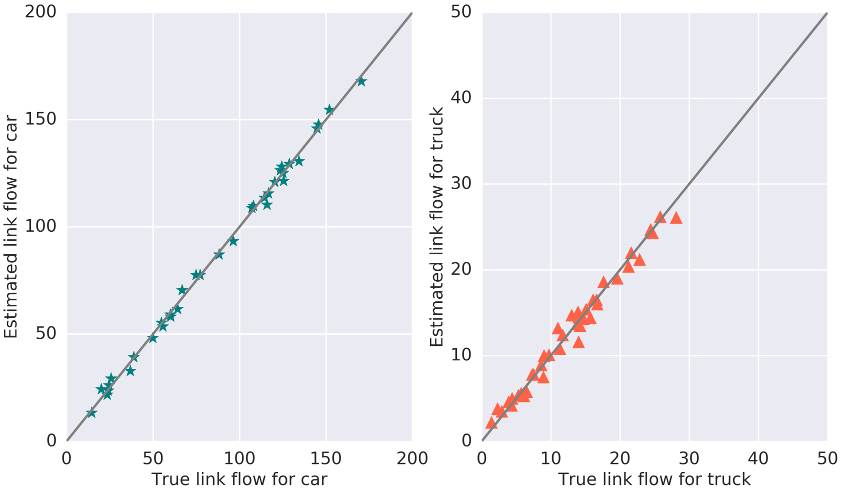

Figure 8: Estimated and "true" link flow for cars and trucks (unit:vehicle/15mins).

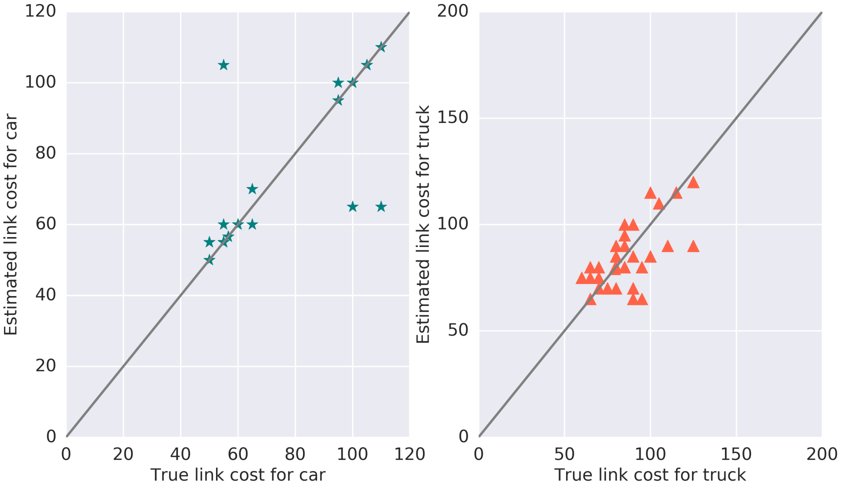

Figure 9: Estimated and "true" link travel time for cars and trucks (unit:seconds).

4.1.3. Multiprocessing

In this section, we demonstrate the power of multiprocessing in solving the MCDODE framework. We solve the baseline setting four times using different number of processes. We use the delayed version of Adagrad to enable the multiprocessing (Zinkevich et al. 2009). We examine the convergence curves for 1, 2, 4, and 8 processes and plot the results in Figure 12. We note that different from previous figures, the x-axis in Figure 12 is the time rather than iterations.

All four methods converges to the same optimal solution. One can see that the 1-process method converges in 50s, 2-process method converges in 30s, and the 4-process method converges in 20 ∼ 25s. By using the multiprocessing, the solving time for the MCDODE framework can be reduced by at least a half. The marginal effects of adding processes decreases when the number of processes increases, as a result of the increasing communication costs among different processes. Since we conduct the experiments on an eight-core CPU, the eight-process method makes use of all computing resources of the CPU and hence it achieves the best convergence rate.

4.1.4. Impact of data quantity

In this section, we analyze the impact of number of data samples on the MCDODE framework. We solve for the baseline setting, while the number of data samples varies from 1 to 256. We keep track of the convergence

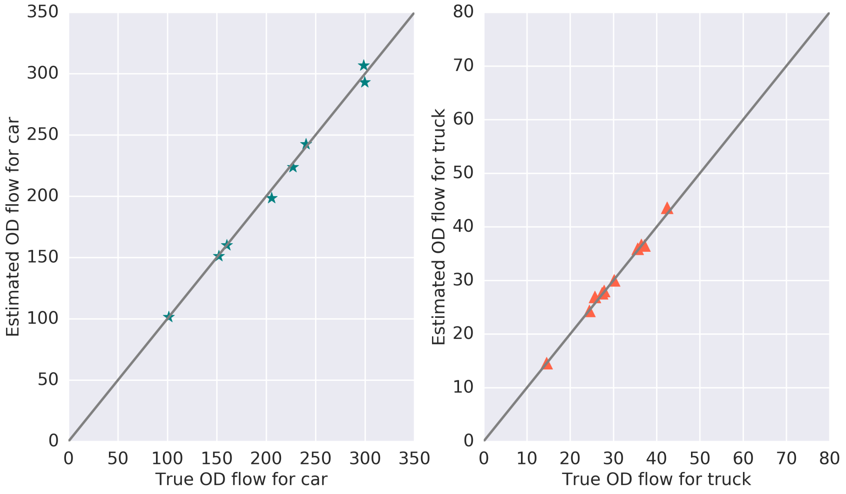

Figure 10: Estimated and "true" OD demand for cars and trucks (unit:vehicle/15mins).

	Car	Truck
Observed flow	0.9992	0.9858
Link flow	0.9982	0.9808
Link travel time	0.9309	0.9586
OD demand	0.9965	0.9940

Table 4: R-square between the "true" and estimated flow and travel time for cars and trucks

curves for different number of data samples, the results are plotted in Figure 13.

As can be seen from Figure 13, the convergence rate increases when the number of data sample increases. The solution method with one data sample does not converge in 100 iterations while the solution method with 256 data samples converges within 10 iterations. In the large-scale networks, limited number of iterations can be conducted due to the lack of computational resources and time constraints, hence more data samples are usually required to ensure estimation accuracy.

4.1.5. Impact of noise level

We further demonstrate the impact of noise level on the proposed solution algorithm. We solve the MCDODE framework for the baseline setting, while we change the noise level from 0 to 0.9. The convergence curves under different noise levels are presented in Figure 14.

One can see from Figure 14 that the noise level has a significant impact on the estimation accuracy of the multi-class OD demand. When there is no noise in the observed data, the estimation method can converges to nearly zero loss in 30 iterations. When the noise level is high, the estimation method does not converge to the optimal solution. Especially when noise level is 0.9, the loss can only be reduced by half and the R-square for the estimated OD is around 0.7.

4.1.6. Sensitivity analysis

In this section, we conduct the sensitivity analysis of the proposed MCDODE framework in terms of initial OD demand, "true" OD demand and step sizes.

Firstly, we perform the sensitivity analysis on the initial OD demand. We solve the MCDODE framework for the baseline setting for 100 times. In each time, we change the initial OD demand by random generators $\texttt{Unif}(0, 15)$ for car demands and $\texttt{Unif}(0, 3)$ for truck demands. We compute the R-squares of the estimated OD demand for all 100 estimations and the boxplot for the R-squares is presented in Figure 15.

As can be seen from Figure 15, the R-squares for car flow are above 0.98 and the variance is low for different initial OD demand, while the variance of R-squares for truck flow is high. Especially for link flow, the R-square

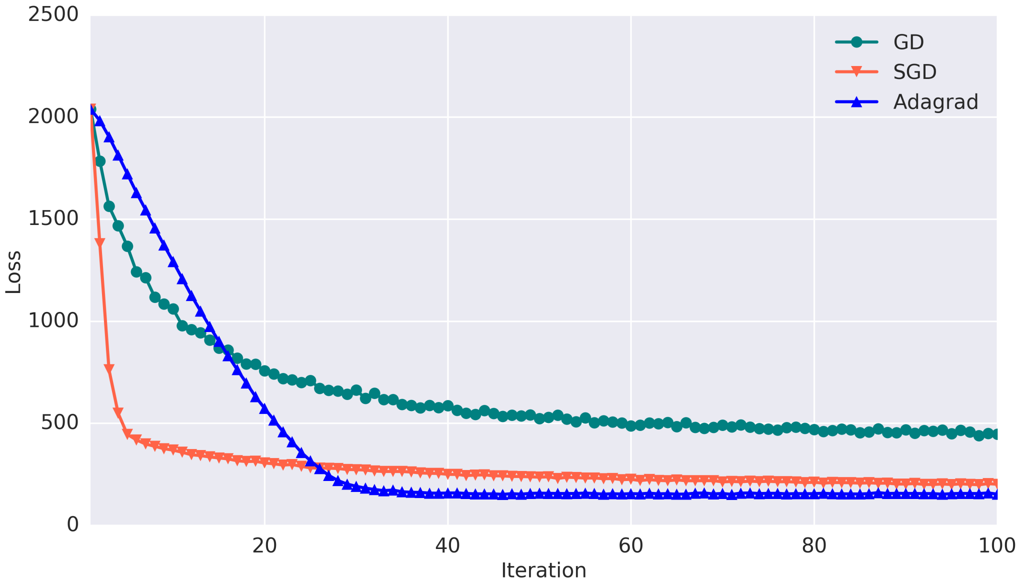

Figure 11: Comparison among GD, SGD and Adagrad

between the "true" and estimated link flow for truck can be as low as 0.95. However, all the R-squares for truck flow are still above 0.9. The results imply that the proposed method is generally robust to the initial OD demand, but estimating the truck demand is more challenging than estimating the car demand. To ensure a satisfactory estimation result, it is desirable to run the MCDODE framework multiple times and chose the best one as the final OD demand. In contrast, the R-square of link travel time for car is lower than that for trucks, which is probably because car speeds can vary in a wide range but truck speed is relatively stable.

Secondly, we fix the initial OD demand and solve for the baseline setting for 100 times. In each run, we sample the "true" OD demand from $\texttt{Unif}(0, 300)$ and $\texttt{Unif}(0, 60)$ for car and truck, respectively. We compute the R-squares of estimated OD demand for the 100 runs and the boxplot for the R-squares is presented in Figure 16.

The MCDODE framework achieves satisfactory accuracy for different "true" OD demands, and most of the R-squares for both car and truck floware over 0.9. The R-square for truck flow is lower than that for car flow, and the variance of R-squares for truck flow is also higher. As we discussed above, estimating the truck OD demand is more challenging, as a result of the discretized and stochastic behaviors of trucks in the traffic simulation model. In addition, the R-squares of link travel time for cars are lower than that for trucks, which is similar to previous study for the initial OD demand.

Thirdly, we examine the Adagrad method with different step sizes. We solve the MCDODE framework for 7 times under the baseline setting. In each run, we vary the step size from 0.01 to 100, and the convergence curves are presented in Figure 17.

One can see that the Adagrad is robust to the step size, and any step size between 0.5 and 2 can guarantee the method to converge to the optimal solution within 60 iterations. When the step size is too small, the method converges slowly; when the step is too large, the convergence becomes unstable and fluctuating.

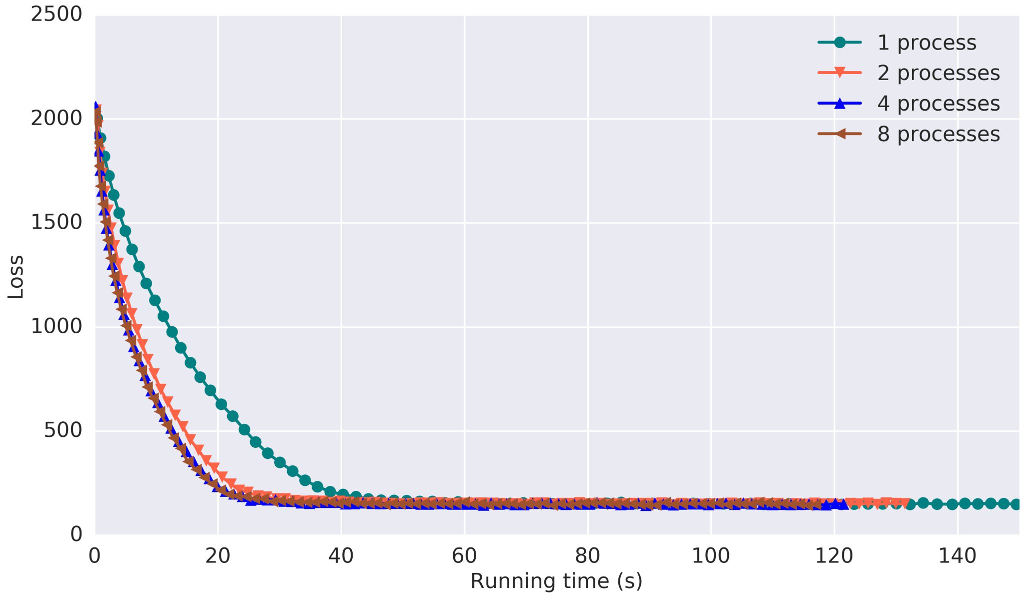

Figure 12: Convergence curves using 1, 2, 4, 8 processes

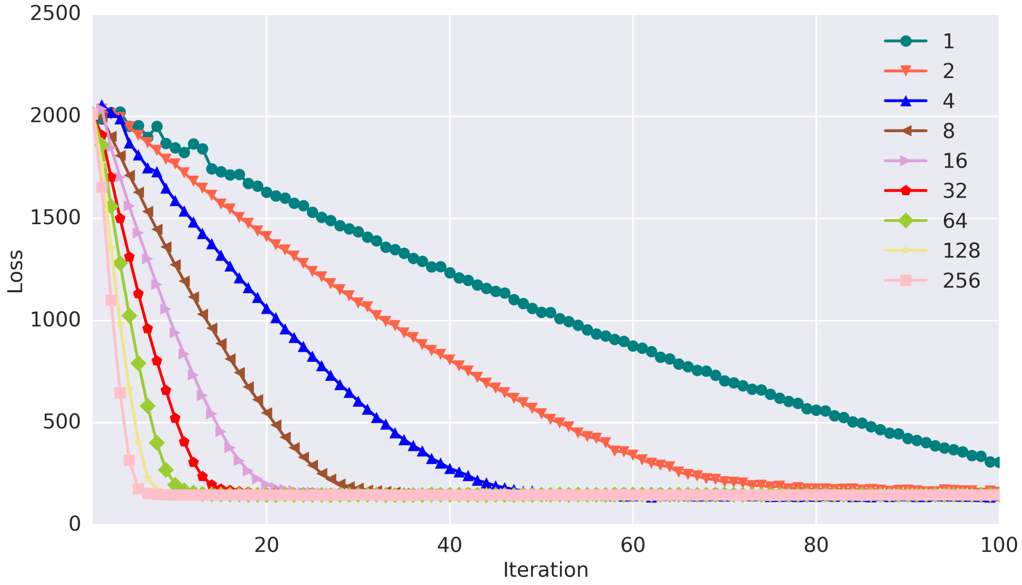

Figure 13: Convergence curves with different number of data samples

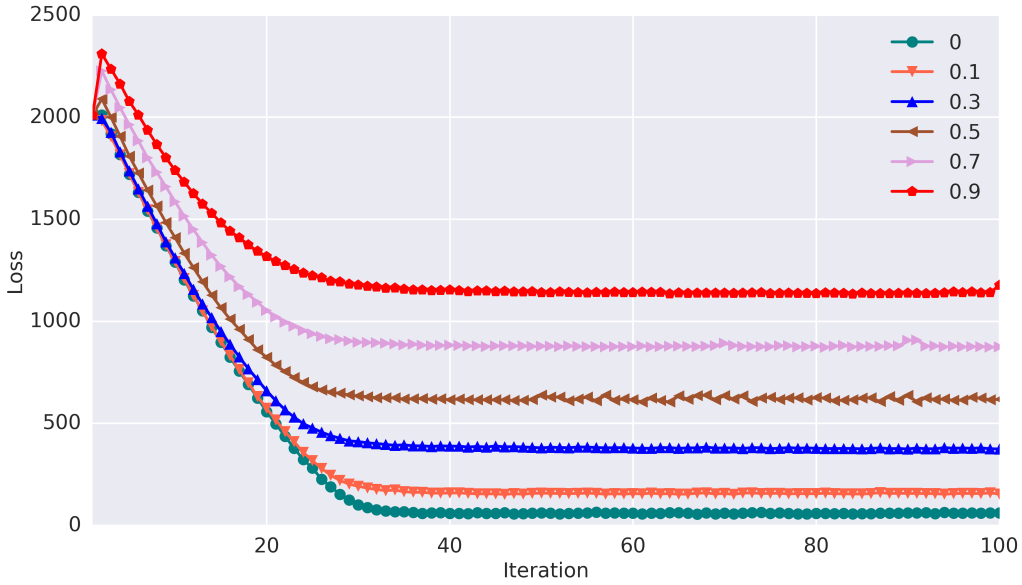

Figure 14: Convergence curves under different noise levels

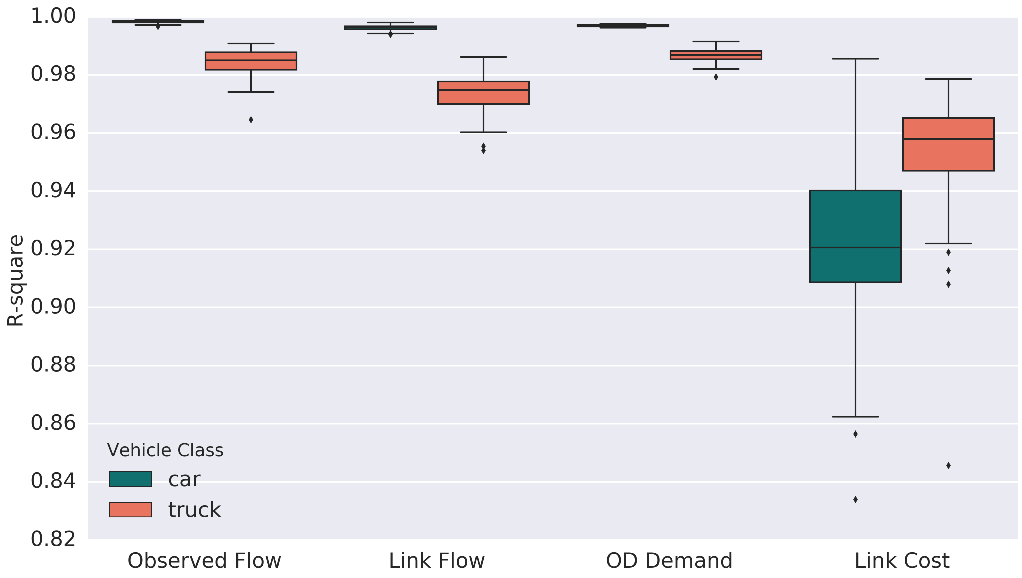

Figure 15: Boxplot for R-squares with different initial OD demand.

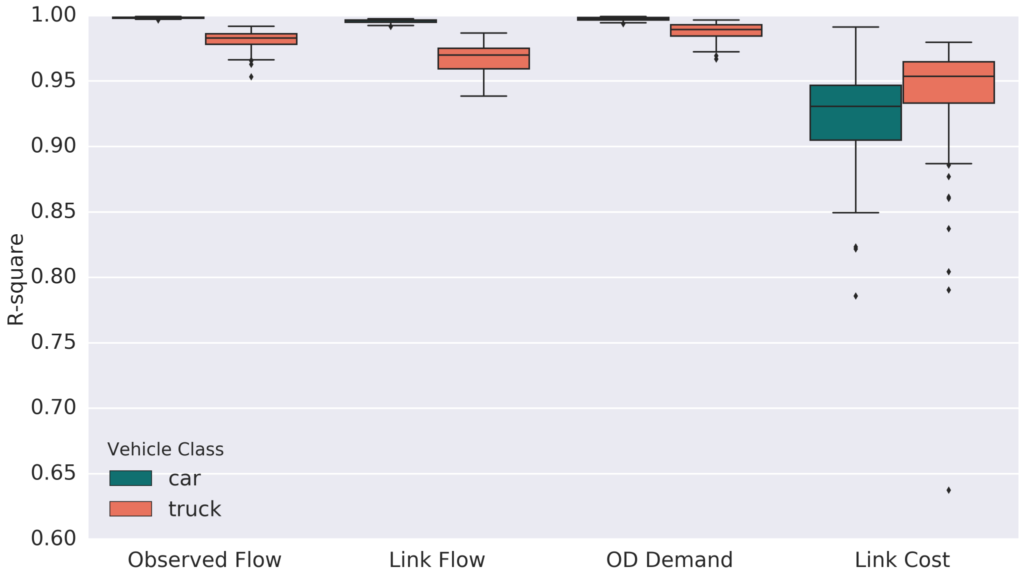

Figure 16: Boxplot for R-squares with different “true” OD demand.

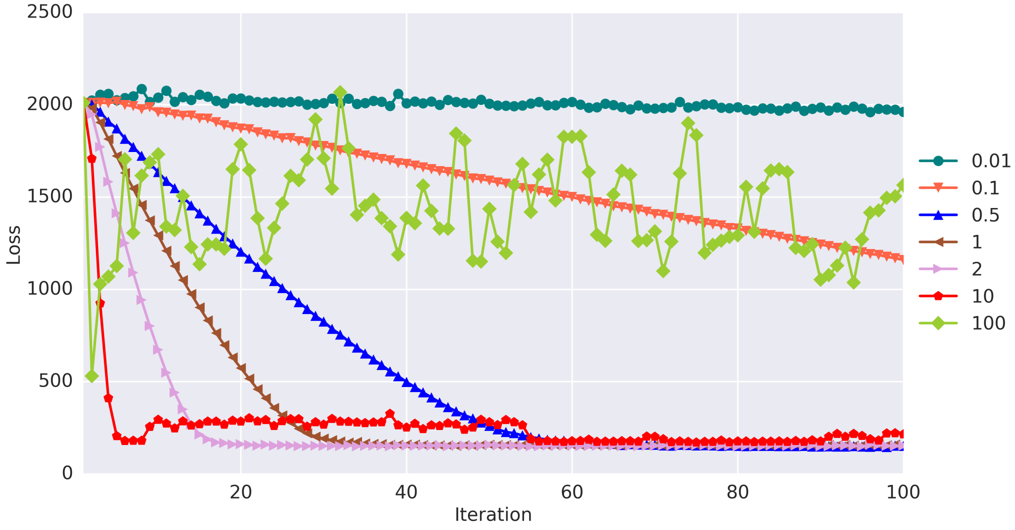

Figure 17: Convergence curves with different step sizes

4.2. A large-scale network: southwestern Pennsylvania region

In this section, we perform the MCDODE on a large-scale network for the southwestern Pennsylvania region (Figure 18). The network covers ten counties of southwestern Pennsylvania region, with the Pittsburgh city located in the center. The approximate range of the 10 counties are marked by the black quadrangle in Figure 18. There are around 2.57 million population and 7,112 square miles area in the network. All parameters for the Pittsburgh network are listed in the Table 5.

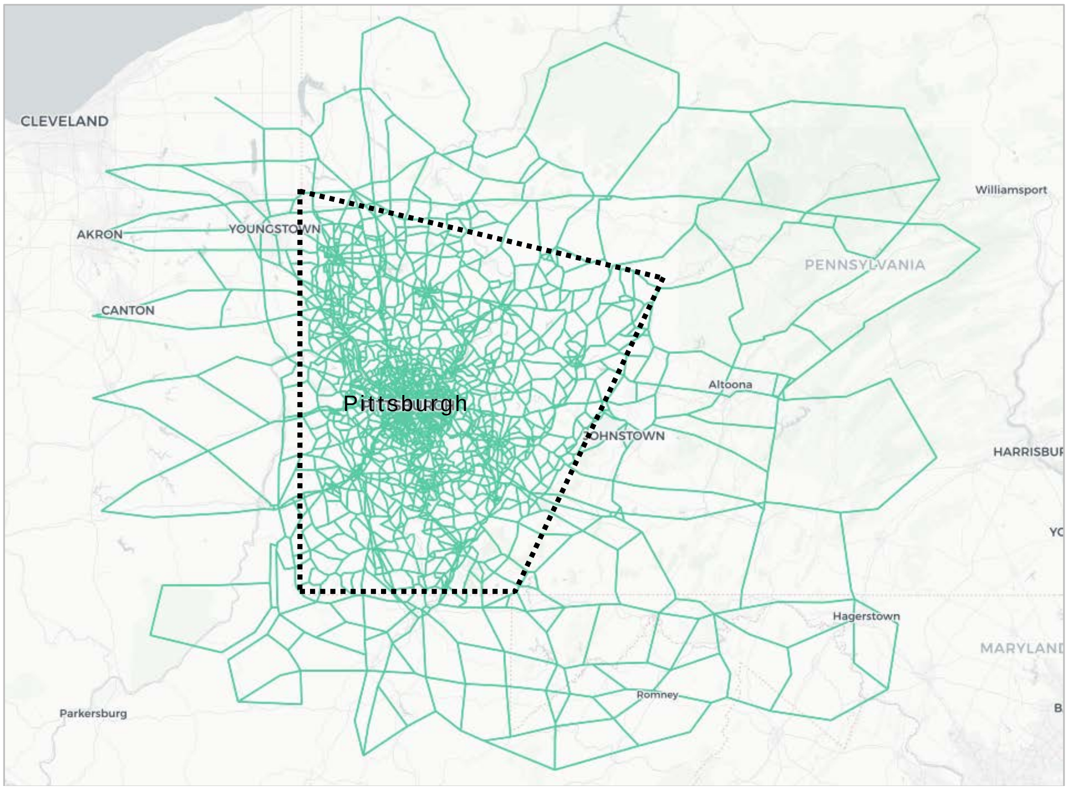

Figure 18: An overview of the network for the southwestern Pennsylvania region

Table 5: Network parameters

Name	Value
studying period (weekday)	6:00 AM - 11:00 AM
simulation unit time	5 seconds
Length of time interval	15 minutes
Number of time intervals	30
number of links	16,110
number of nodes	6,297
number of origins (destinations)	283
number of origin-destination pairs	80,089

We run the MCDODE framework with traffic flow data and travel time (traffic speed) data. The traffic flow data are obtained from the Pennsylvania Department of Transportation (PennDOT). The data are collected annually for some selected locations on the Pennsylvania state-owned roads, for each hour of the day and for one day of the year. The car traffic volume counts and truck traffic volume counts are collected separately, where car traffic volume counts are measured for all passenger cars and truck traffic volume counts includes all kinds of trucks at the measured location. The traffic count can be either one-directional or bi-directional. Since all count data are measured in hours, in data pre-processing we smooth the hourly count data to 15-minute interval. There are 608 locations in total that has valid car and truck volume counts for MCDODE. Traffic speed data are obtained Federal Highway Administration (FHWA) for the year 2016. The speed data are observed every 5-minutes of the day for highway segments, the data are also classified to cars and trucks. We aggregate the travel time data to 15-minute interval. In total, there are 945 locations with valid car and truck travel time observations.

The initial OD demands for cars and trucks are randomly generated from a uniform distribution $\texttt{Unif}(0, 0.01)$ and $\texttt{Unif}(0, 0.001)$, respectively. We aggregate all the traffic flow and travel time observations to a single data sample and use the single-process Adagrad method to solve MCDODE. The step size is e^{-5}. We set $w_1 = 1, w_2 = 0.01$. We use the hybrid dynamic traffic assignment model as the route choice model (Qian & Zhang 2013), the adaptive ratio is 0.2 for cars and 0 for trucks. The MCDODE framework runs for 55 iterations. In first 35 iterations, both OD demand for car and trucks are updated simultaneously; while for the last 20 iterations, we only update the truck demand since its more challenging to estimate.

The convergence of the proposed MCDODE framework is presented in Figure 19. Overall, the solution algorithm performs well and the objective function converges fast. It takes around $25 \times 55 = 1375$ minutes (around 23 hours) to complete the 55 iterations. For each iteration, the traffic simulation takes around 7 minutes, and the other 18 minutes are used for the demand estimation. Constructing DAR matrix is identified as the bottleneck in the demand estimation.

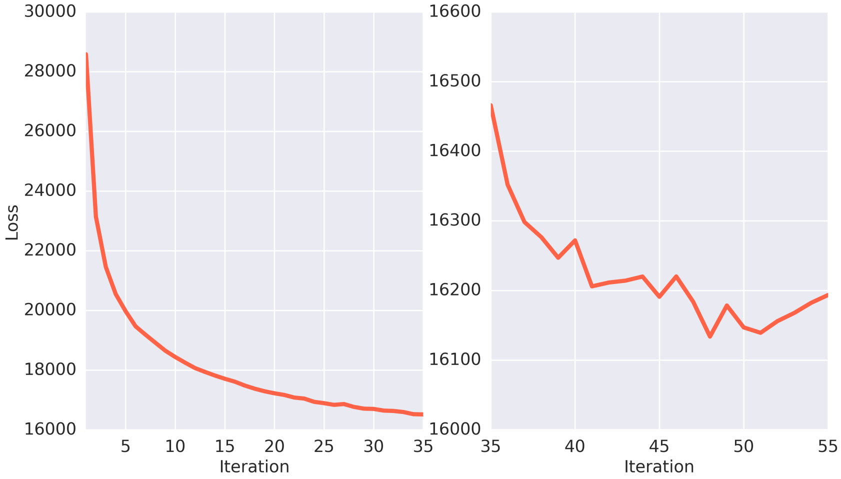

Figure 19: Convergence of the objective function for 55 iterations (OD demands for cars and trucks are updated simultaneously in first 35 iterations, while only truck OD demand is updated in last 20 iterations.)

The comparisons between the observed and estimated flow are presented in Figure 20, and R-squares are 0.66 and 0.59 for car and truck, respectively. One can see that the truck flow is roughly one tenth of the car flow, and the estimation accuracy for car flow is higher than truck flow. Overall, the results of the MCDODE framework is compelling and satisfactory for such a large network.

We also plot the comparisons between the observed and estimated link speed in Figure 21. The reason we plot the link speed instead of link travel time is that the link travel time can vary in a wide range for large scale networks, but link speed usually varies between 20 to 80 miles/hour. Hence, visualizing the link speed is more straightforward and legible.

The R-squares between the observed and estimated link speed are 0.45 and 0.51 for car and truck, respectively. One can see that the estimation accuracy of traffic speed is not as good as the traffic flow, which is probably attributed to unsatisfactory approximations of the link travel time derivatives $\frac{\partial \bar{\Lambda}(\{\mathbf{X}_i\}_i)}{\partial \mathbf{X}_i}$ in dynamic networks. Improving the approximation quality for the link travel time will be left for the future research.

In addition, the estimation uses the randomly generated OD demand as the initial point for the MCDODE framework. If prior knowledge of the OD demand can be obtained from traditional planning models, the estimation accuracy might further be improved. The estimated multi-class OD demand is further used to study the impact of a development project in Pittsburgh, and details are presented in Pi et al. (2018).

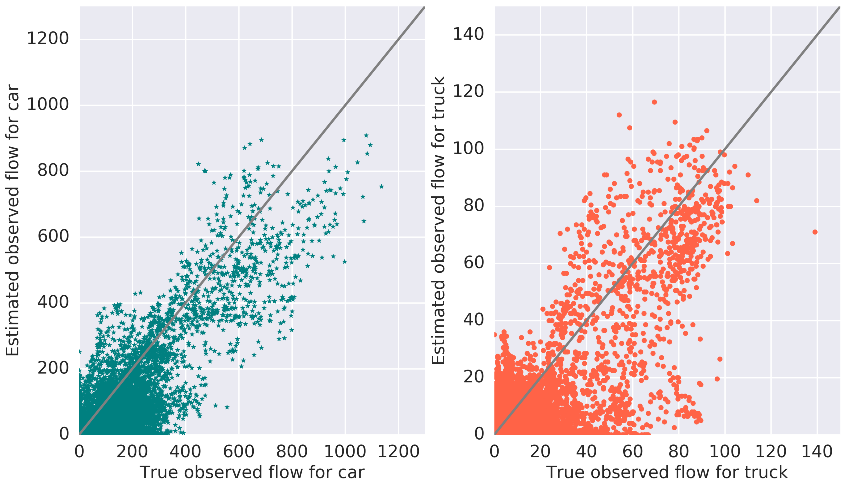

Figure 20: Estimated and observed flow for cars and trucks (unit:vehicle/15mins)

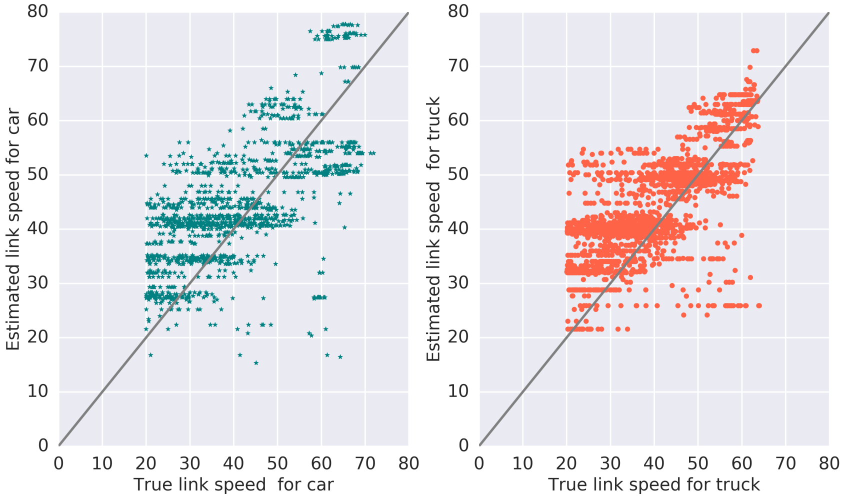

Figure 21: Estimated and observed flow for cars and trucks (unit:miles/hour)

5. Conclusion

This paper presents a data-driven framework for multi-class dynamic origin-destination demand estimation (MCDODE) using observed traffic flow and travel time data. The traffic data can be any linear combinations of flow characteristics (e.g. counts, time or travel time) across vehicle classes, road segments and time intervals. All the characteristics involved in the MCDODE formulation are vectorized, and the proposed framework is represented on a computational graph. The computational graph can be solved efficiently through a forward-backward algorithm for large-scale MCDODE problems. In the forward iteration, the dynamic traffic assignment problem is solved, and the loss (objective function) is computed through a series of equations. In the backward iteration, the OD demand is updated by the backpropagation method with the route choice matrix, DAR matrix and route travel time known from the forward iteration. The MCDODE formulation is solved when the forward-backward algorithm converges.

Practical issues related to MCDODE framework are discussed. We adopt a mesoscopic multi-class traffic simulation package MAC-POSTS to solve for the spatio-temporal path/link flow. The DAR matrix is highly sparse, and thus we propose novel tree-based cumulative curves from MAC-POSTS to construct the sparse DAR matrix. We incorporate multi-day observation data to the MCDODE framework, and different variants of gradient-based solution algorithms are discussed and compared.

The proposed MCDODE framework is examined on a small network as well as a real-world large-scale network. The objective function converges quickly with the Adagrad method. We also conduct the sensitivity analysis of the estimation accuracy with respect to initial OD demand, "true" OD demand, and step sizes. Overall, the estimation results are compelling, satisfactory and robust, and the forward-backward algorithm is computationally plausible for large-scale networks. The estimated multi-class dynamic OD demand can help policymakers to better understand the dynamics of OD demand and the spatio-temporal distribution of vehicles in terms of different classes.

In the near future, we plan to improve the estimation accuracy of the MCDODE framework in the following two directions: 1) some prior knowledge about the dynamic OD demand can be used as the initial point for the solution methods. For example, we can construct the DAR matrix from the speed data directly and estimate the dynamic OD without running the simulation. The estimated OD demand can be used as the initial point for the proposed MCDODE framework (Ma & Qian 2018*b*); 2) the method of approximating the derivatives of link travel time under multi-class flow can be further improved. In addition, we plan to extend this research to calibrate the parameters in route choice models as well as the fundamental diagrams, thanks to the versatile computational graph framework. We also plan to extend this research to estimate the probabilistic distribution of multi-class dynamic OD demand and explore the spatio-temporal characteristics of dynamic OD demand.

Supplementary materials

The mesoscopic multi-class traffic simulation package MAC-POSTS [1] and the MCDODE framework [2] are implemented and open-sourced on Github.

Acknowledgments

This research is supported by NSF Award CMMI-1751448, and Carnegie Mellon University's Mobility 21, a National University Transportation Center for Mobility sponsored by the US Department of Transportation.

[1] https://github.com/Lemma1/MAC-POSTS

[2] https://github.com/Lemma1/MAC-POSTS/tree/master/src/examples/mcDODE

References

Antoniou, C., Azevedo, C. L., Lu, L., Pereira, F. & Ben-Akiva, M. (2015), 'W-spsa in practice: Approximation of weight matrices and calibration of traffic simulation models', *Transportation Research Part C: Emerging Technologies* **59**, 129–146.

Antoniou, C., Barceló, J., Breen, M., Bullejos, M., Casas, J., Cipriani, E., Ciuffo, B., Djukic, T., Hoogendoorn, S., Marzano, V. et al. (2015), 'Towards a generic benchmarking platform for origin–destination flows estimation/updating algorithms: Design, demonstration and validation', *Transportation Research Part C: Emerging Technologies* .

Ashok, K. & Ben-Akiva, M. E. (2000), 'Alternative approaches for real-time estimation and prediction of time-dependent origin–destination flows', *Transportation Science* **34**(1), 21–36.

Balakrishna, R., Ben-Akiva, M. & Koutsopoulos, H. (2008), Time-dependent origin-destination estimation without assignment matrices, *in* 'Second International Symposium of Transport Simulation (ISTS06). Lausanne, Switzerland. 4-6 September 2006', EPFL Press.

Ben-Akiva, M., Bierlaire, M., Koutsopoulos, H. & Mishalani, R. (1998), Dynamit: a simulation-based system for traffic prediction, *in* 'DACCORD Short Term Forecasting Workshop', pp. 1–12.

Ben-Akiva, M. E., Gao, S., Wei, Z. & Wen, Y. (2012), 'A dynamic traffic assignment model for highly congested urban networks', *Transportation research part C: emerging technologies* **24**, 62–82.

Bierlaire, M. & Crittin, F. (2004), 'An efficient algorithm for real-time estimation and prediction of dynamic od tables', *Operations Research* **52**(1), 116–127.

Cascetta, E., Inaudi, D. & Marquis, G. (1993), 'Dynamic estimators of origin-destination matrices using traffic counts', *Transportation science* **27**(4), 363–373.

Cipriani, E., Florian, M., Mahut, M. & Nigro, M. (2011), 'A gradient approximation approach for adjusting temporal origin–destination matrices', *Transportation Research Part C: Emerging Technologies* **19**(2), 270–282.

Dafermos, S. C. (1972), 'The traffic assignment problem for multiclass-user transportation networks', *Transportation science* **6**(1), 73–87.

Davis, G. A. (1994), 'Exact local solution of the continuous network design problem via stochastic user equilibrium assignment', *Transportation Research Part B: Methodological* **28**(1), 61–75.

Duchi, J., Hazan, E. & Singer, Y. (2011), 'Adaptive subgradient methods for online learning and stochastic optimization', *Journal of Machine Learning Research* **12**(Jul), 2121–2159.

Fisk, C. (1989), 'Trip matrix estimation from link traffic counts: the congested network case', *Transportation Research Part B: Methodological* **23**(5), 331–336.

Florian, M. & Chen, Y. (1995), 'A coordinate descent method for the bi-level o–d matrix adjustment problem', *International Transactions in Operational Research* **2**(2), 165–179.

Gal, Y. & Ghahramani, Z. (2016), Dropout as a bayesian approximation: Representing model uncertainty in deep learning, *in* 'international conference on machine learning', pp. 1050–1059.

Gundaliya, P., Mathew, T. V. & Dhingra, S. L. (2008), 'Heterogeneous traffic flow modelling for an arterial using grid based approach', *Journal of Advanced Transportation* **42**(4), 467–491.

Hazelton, M. L. (2008), 'Statistical inference for time varying origin–destination matrices', *Transportation Research Part B: Methodological* **42**(6), 542–552.

Huang, H.-J. & Li, Z.-C. (2007), 'A multiclass, multicriteria logit-based traffic equilibrium assignment model under atis', *European Journal of Operational Research* **176**(3), 1464–1477.

Jha, M., Gopalan, G., Garms, A., Mahanti, B., Toledo, T. & Ben-Akiva, M. (2004), 'Development and calibration of a large-scale microscopic traffic simulation model', *Transportation Research Record: Journal of the Transportation Research Board* (1876), 121–131.

Jin, W.-L. (2012), 'A link queue model of network traffic flow', *arXiv preprint arXiv:1209.2361* .

LeBlanc, L. J. & Farhangian, K. (1982), 'Selection of a trip table which reproduces observed link flows', *Transportation Research Part B: Methodological* **16**(2), 83–88.

Lee, J.-B. & Ozbay, K. (2009), 'New calibration methodology for microscopic traffic simulation using enhanced simultaneous perturbation stochastic approximation approach', *Transportation Research Record: Journal of the Transportation Research Board* (2124), 233–240.

Lu, C.-C., Zhou, X. & Zhang, K. (2013), 'Dynamic origin–destination demand flow estimation under congested traffic conditions', *Transportation Research Part C: Emerging Technologies* **34**, 16–37.

Lu, L., Xu, Y., Antoniou, C. & Ben-Akiva, M. (2015), 'An enhanced spsa algorithm for the calibration of dynamic traffic assignment models', *Transportation Research Part C: Emerging Technologies* **51**, 149–166.

Ma, J., Zhang, H. M. et al. (2008), 'A polymorphic dynamic network loading model', *Computer-Aided Civil and Infrastructure Engineering* **23**(2), 86–103.

Ma, R., Ban, X. J. & Pang, J.-S. (2014), 'Continuous-time dynamic system optimum for single-destination traffic networks with queue spillbacks', *Transportation Research Part B: Methodological* **68**, 98–122.

Ma, W., Pinchao, Z. & Qian, Z. (2016), 'Dynamic network analysis & real-time traffic management for philadelphia metropolitan area'.

Ma, W. & Qian, Z. (2018*a*), 'A generalized single-level formulation for origin–destination estimation under stochastic user equilibrium', *Transportation Research Record* p. 0361198118782041.

Ma, W. & Qian, Z. S. (2018*b*), 'Estimating multi-year 24/7 origin-destination demand using high-granular multi-source traffic data', *Transportation Research Part C: Emerging Technologies* **96**, 96–121.

Maher, M. J., Zhang, X. & Van Vliet, D. (2001), 'A bi-level programming approach for trip matrix estimation and traffic control problems with stochastic user equilibrium link flows', *Transportation Research Part B: Methodological* **35**(1), 23–40.

Mahmassani, H. & Herman, R. (1984), 'Dynamic user equilibrium departure time and route choice on idealized traffic arterials', *Transportation Science* **18**(4), 362–384.

Mahmassani, H., Hu, T. & Jayakrishnan, R. (1992), Dynamic traffic assignment and simulation for advanced network informatics (dynasmart), *in* 'Proceedings of the 2nd international CAPRI seminar on Urban Traffic Networks, Capri, Italy'.

Neyshabur, B., Tomioka, R. & Srebro, N. (2014), 'In search of the real inductive bias: On the role of implicit regularization in deep learning', *arXiv preprint arXiv:1412.6614* .

Nguyen, S. (1977), *Estimating and OD Matrix from Network Data: a Network Equilibrium Approach*, Montréal: Université de Montréal, Centre de recherche sur les transports.

Nie, Y. M. & Zhang, H. M. (2008), 'A variational inequality formulation for inferring dynamic origin–destination travel demands', *Transportation Research Part B: Methodological* **42**(7), 635–662.

Nie, Y. M. & Zhang, H. M. (2010), 'Solving the dynamic user optimal assignment problem considering queue spillback', *Networks and Spatial Economics* **10**(1), 49–71.

Noriega, Y. & Florian, M. (2007), 'Multi-class demand matrix adjustment'.

Osorio, C. (2019), 'Dynamic origin-destination matrix calibration for large-scale network simulators', *Transportation Research Part C: Emerging Technologies* **98**, 186–206.

Pan, S. J. & Yang, Q. (2010), 'A survey on transfer learning', *IEEE Transactions on knowledge and data engineering* **22**(10), 1345–1359.

Peeta, S. & Ziliaskopoulos, A. K. (2001), 'Foundations of dynamic traffic assignment: The past, the present and the future', *Networks and spatial economics* **1**(3-4), 233–265.

Pi, X., Ma, W. & Qian, Z. S. (2018), Regional traffic planning and operation using mesoscopic car-truck traffic simulation.

Prashker, J. N. & Bekhor, S. (2004), 'Route choice models used in the stochastic user equilibrium problem: a review', *Transport reviews* **24**(4), 437–463.

Qian, Z. S., Li, J., Li, X., Zhang, M. & Wang, H. (2017), 'Modeling heterogeneous traffic flow: A pragmatic approach', *Transportation Research Part B: Methodological* **99**, 183–204.

Qian, Z. S., Shen, W. & Zhang, H. (2012), 'System-optimal dynamic traffic assignment with and without queue spillback: Its path-based formulation and solution via approximate path marginal cost', *Transportation research part B: methodological* **46**(7), 874–893.

Qian, Z. S. & Zhang, H. M. (2013), 'A hybrid route choice model for dynamic traffic assignment', *Networks and Spatial Economics* **13**(2), 183–203.

Raothanachonkun, P., Sano, K. & Matsumoto, S. (2006), 'Estimation of multiclass origin-destination matrices using genetic algorithm', *Infrastructure Planning Review* **23**, 423–432.

Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1985), Learning internal representations by error propagation, Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

Saad, D. (1998), 'Online algorithms and stochastic approximations', *Online Learning* **5**.

Shao, H., Lam, W. H., Sumalee, A. & Hazelton, M. L. (2015), 'Estimation of mean and covariance of stochastic multi-class od demands from classified traffic counts', *Transportation Research Part C: Emerging Technologies* .

Shen, W., Nie, Y. & Zhang, H. M. (2007), On path marginal cost analysis and its relation to dynamic system-optimal traffic assignment, *in* 'Transportation and Traffic Theory 2007. Papers Selected for Presentation at ISTTT17'.

Tavana, H. (2001), 'Internally-consistent estimation of dynamic network origin-destination flows from intelligent transportation systems data using bi-level optimization'.

Tympakianaki, A., Koutsopoulos, H. N. & Jenelius, E. (2015), 'c-spsa: Cluster-wise simultaneous perturbation stochastic approximation algorithm and its application to dynamic origin–destination matrix estimation', *Transportation Research Part C: Emerging Technologies* **55**, 231–245.

Vaze, V., Antoniou, C., Wen, Y. & Ben-Akiva, M. (2009), 'Calibration of dynamic traffic assignment models with point-to-point traffic surveillance', *Transportation Research Record: Journal of the Transportation Research Board* (2090), 1–9.

Venkatesan, K., Gowri, A. & Sivanandan, R. (2008), 'Development of microscopic simulation model for heterogeneous traffic using object oriented approach', *Transportmetrica* **4**(3), 227–247.

Wong, S., Tong, C., Wong, K., Lam, W. K., Lo, H., Yang, H. & Lo, H. (2005), 'Estimation of multiclass origin-destination matrices from traffic counts', *Journal of urban planning and development* **131**(1), 19–29.

Wu, X., Guo, J., Xian, K. & Zhou, X. (2018), ‘Hierarchical travel demand estimation using multiple data sources: A forward and backward propagation algorithmic framework on a layered computational graph’, *Transportation Research Part C: Emerging Technologies* **96**, 321–346.

Yang, H. (1995), ‘Heuristic algorithms for the bilevel origin-destination matrix estimation problem’, *Transportation Research Part B: Methodological* **29**(4), 231–242.

Yang, H. & Huang, H.-J. (2004), ‘The multi-class, multi-criteria traffic network equilibrium and systems optimum problem’, *Transportation Research Part B: Methodological* **38**(1), 1–15.

Yang, H., Sasaki, T., Iida, Y. & Asakura, Y. (1992), ‘Estimation of origin-destination matrices from link traffic counts on congested networks’, *Transportation Research Part B: Methodological* **26**(6), 417–434.

Yang, Y., Fan, Y. & Wets, R. J. (2018), ‘Stochastic travel demand estimation: Improving network identifiability using multi-day observation sets’, *Transportation Research Part B: Methodological* **107**, 192–211.

Zhang, C., Osorio, C. & Flötteröd, G. (2017), ‘Efficient calibration techniques for large-scale traffic simulators’, *Transportation Research Part B: Methodological* **97**, 214–239.

Zhang, H., Nie, Y. & Qian, Z. (2008), ‘Estimating time-dependent freeway origin-destination demands with different data coverage: Sensitivity analysis’, *Transportation Research Record: Journal of the Transportation Research Board* (2047), 91–99.

Zhang, H., Nie, Y. & Qian, Z. (2013), ‘Modelling network flow with and without link interactions: the cases of point queue, spatial queue and cell transmission model’, *Transportmetrica B: Transport Dynamics* **1**(1), 33–51.

Zhao, Q., Turnquist, M. A., Dong, Z. & He, X. (2018), ‘Multiclass probit-based origin–destination estimation using multiple data types’, *Journal of Transportation Engineering, Part A: Systems* **144**(6), 04018018.

Zhou, X. & Mahmassani, H. S. (2007), ‘A structural state space model for real-time traffic origin–destination demand estimation and prediction in a day-to-day learning framework’, *Transportation Research Part B: Methodological* **41**(8), 823–840.

Zhou, X., Qin, X. & Mahmassani, H. (2003), ‘Dynamic origin-destination demand estimation with multiday link traffic counts for planning applications’, *Transportation Research Record: Journal of the Transportation Research Board* (1831), 30–38.

Zhou, X. & Taylor, J. (2014), ‘Dtalite: A queue-based mesoscopic traffic simulator for fast model evaluation and calibration’, *Cogent Engineering* **1**(1), 961345.

Zhou, Z., Chen, A. & Bekhor, S. (2012), ‘C-logit stochastic user equilibrium model: formulations and solution algorithm’, *Transportmetrica* **8**(1), 17–41.

Zinkevich, M., Langford, J. & Smola, A. J. (2009), Slow learners are fast, *in* ‘Advances in neural information processing systems’, pp. 2331–2339.