Joint Data-Driven Recovery of Travel Time Functions and Origin-Destination Demand in Multi-Class Transportation Networks[1]

Jing Zhang[a], Ioannis Ch. Paschalidis[b,*]

[a]*Division of Systems Eng., Boston University, Brookline, MA 02446.*
[b]*Dept. of Electrical and Computer Eng., Boston University, 8 St. Mary's St., Boston, MA 02215.*

Abstract

For both single-class and multi-class transportation networks, existing works have tackled the problem of adjusting Origin-Destination (OD) demand matrices and recovering travel latency cost functions. However, these two types of problems are typically treated separately. In this paper, we propose a method to jointly recover nonparametric travel latency cost functions and adjust OD demand matrices. We formulate the problem as a bilevel optimization problem in a multi-class transportation network and develop an alternating optimization approach to solve it. Extensive numerical experiments using benchmark networks and a real road network (the Eastern Massachusetts highway network), ranging from moderate-sized to large-sized, demonstrate the effectiveness and efficiency of our method.

Keywords:

1. Introduction

Depending on whether we put different weights onto the flows from different classes of vehicles, we can model a transportation network as *single-class* [1] or *multi-class* [2]. Naturally, we can treat the former as a special case of the latter. Given a general multi-class transportation network and assuming all users (drivers) choose routes selfishly, the network reaches an equilibrium in terms of link flows, known as the *Wardrop equilibrium* [3]. At the equilibrium, no single driver can "benefit" by rerouting.

Mathematically, for a general multi-class transportation network, given *travel latency (time) cost functions* (we will also simply say *cost functions*) with respect to link (road/arc) flows and an Origin-Destination (OD) flow demand matrix, finding the Wardrop equilibrium is formulated as the Traffic Assignment Problem (TAP), which has been extensively studied; see, e.g., [4] and the references therein. The TAP, which we will

[1]Research partially supported by the NSF under grants CNS-1645681, CCF-1527292 and IIS-1237022 and by the ARO under grant W911NF-12-1-0390.

*Corresponding author.
Email addresses: jzh@bu.edu (Jing Zhang), yannisp@bu.edu (Ioannis Ch. Paschalidis)
URL: http://sites.bu.edu/paschalidis/ (Ioannis Ch. Paschalidis)

call the *forward problem* throughout the paper, can be explicitly formulated as an optimization problem [1] or a Variational Inequality (VI) problem [5].

In practice, however, the OD demand matrix and the cost function are not readily available. Transportation engineers have been typically using surveys to estimate demand and making empirical assumptions for the cost function (e.g., Bureau of Public Roads cost function [6]). With the increasing availability of data, there is an opportunity to obtain these quantities in a more systematic, data-driven fashion by formulating appropriate *inverse problems*. More specifically, given an OD demand matrix and the Wardrop equilibrium, recovering the cost function has been recently studied for single-class networks [7, 8, 9]. Our recent work [10], a preliminary version of the current paper, extends the same type of inverse problem to multi-class models but without considering them jointly with demand estimation. At the same time, another version of the inverse problem which adjusts the OD demand matrix given the Wardrop equilibrium and the cost function has also been studied for both single-class [11, 12, 9, 13] and multi-class [14] networks. For convenience, in the sequel, we use the term IP-1 (resp., IP-2) to indicate the inverse problem of recovering cost functions (resp., adjusting OD demand matrices).

Most of the existing work typically deals with these two types of inverse problems separately; a limitation we seek to address in this paper. There has been some work to estimate OD demand in a congested network, leading to bilevel formulations and local optimization methods, see, e.g., [15, 16]. Closer to the goal of our work, [17] considered the simultaneous estimation of travel cost and OD demand in a stochastic user equilibrium setting. Yet, these earlier works consider single-class networks (with the exception of [14]) and do not attempt to estimate (nonparametrically) the full structure of the travel cost functions as we do. Rather, they seek to estimate a sensitivity constant that adjusts how a given travel cost function affects route choice probabilities.

It is a well-known fact that, for general multi-class transportation networks, there do not exist reasonable easily verifiable assumptions about the cost functions to ensure the existence and uniqueness of the solution to the *forward problem* (i.e., TAP) [18]. Therefore, for the multi-class inverse problem IP-1, it would be hard to establish rigorous theoretical results under mild easy-to-check conditions. In our preliminary work [10], an appropriate formulation for the multi-class IP-1 is provided and numerical experiments are conducted to empirically validate the proposed solution.

In this paper, operating on multi-class transportation networks, we seek to jointly investigate the two related inverse problems – recovering cost functions (IP-1) in a non-parametric setting and adjusting OD demand matrices (IP-2). For a given multi-class network, assume that the Wardrop equilibrium is observed and that an initial "rough" OD demand matrix is available. To solve the joint problem, we leverage a generalized bilevel optimization problem formulation and propose a gradient-based alternating optimization approach. The approach alternates between estimating the cost function (given demand) and adjusting the OD demand matrix (given the cost function). To validate its effectiveness and efficiency, we conduct extensive numerical experiments. In particular, we implement our algorithms over three benchmark networks (Sioux-Falls, Berlin-Tiergarten, and Anaheim) ranging from moderate-sized to large-sized and a real large network representing the Eastern Massachusetts (EMA) highway network.

The rest of the paper is organized as follows. In Sec. 2 we present the single-class and multi-class transportation network models, formulate the *forward problem* (TAP), specify the form of the cost functions, and formulate the *inverse problem* IP-1. We formulate the joint problem and propose algorithms to solve it in Sec. 3. Numerical results are shown in Sec. 4. Sec. 5 concludes the paper. Additional experimental details for the EMA highway network are contained in the Appendix.

Notational conventions: All vectors are column vectors. For economy of space, we write $\mathbf{x} = (x_1, \ldots, x_{\dim(\mathbf{x})})$ to denote the column vector $\mathbf{x}$, where $\dim(\mathbf{x})$ is its dimensionality. $\mathbf{x} \geq \mathbf{0}$ (with $\mathbf{0}$ being the zero vector) denotes that all entries of a vector $\mathbf{x}$ are nonnegative. Denote by $\mathbb{R}_+$ the set of all nonnegative real numbers. We use "prime" to denote the transpose of a matrix or vector. Unless otherwise specified, $\|\cdot\|$ denotes the ℓ_2 norm. Let $|\mathcal{D}|$ denote the cardinality of a set $\mathcal{D}$, and $[\![\mathcal{D}]\!]$ the set $\{1, \ldots, |\mathcal{D}|\}$. $A \stackrel{\text{def}}{=} B$ indicates A is defined using B.

2. Preliminaries

2.1. Transportation network model

Let us review the transportation network models. We note that the same single-class model to be presented first has also been adopted in [7, 8, 9].

2.1.1. Single-class transportation network model

Consider a directed graph denoted by $(\mathcal{V}, \mathcal{A})$, where $\mathcal{V}$ denotes the set of nodes and $\mathcal{A}$ the set of links. Assume it is strongly connected. Let $\mathbf{N} \in \{0, 1, -1\}^{|\mathcal{V}| \times |\mathcal{A}|}$ be the node-link incidence matrix, and $\mathbf{e}_a$ the vector with an entry equal to 1 corresponding to link a and all the other entries equal to 0. Let $\mathbf{w} = (w_s, w_t)$ denote an Origin-Destination (OD) pair and $\mathcal{W} = \{\mathbf{w}_i : \mathbf{w}_i = (w_{si}, w_{ti}), \, i \in [\![\mathcal{W}]\!]\}$ the set of all OD pairs. Denote by $d^{\mathbf{w}} \geq 0$ the amount of the flow demand from w_s to w_t. Let $\mathbf{d}^{\mathbf{w}} \in \mathbb{R}^{|\mathcal{V}|}$ be the vector which is all zeros, except for a $-d^{\mathbf{w}}$ in the coordinate corresponding to node w_s and a $d^{\mathbf{w}}$ in the coordinate corresponding to node w_t. Let x_a be the total link flow on link $a \in \mathcal{A}$ and $\mathbf{x}$ the vector of these flows. Let $\mathcal{F}$ be the set of feasible flow vectors defined by

$$\mathcal{F} \stackrel{\text{def}}{=} \Big\{\mathbf{x} : \exists \mathbf{x}^{\mathbf{w}} \in \mathbb{R}_+^{|\mathcal{A}|} \text{ s.t. } \mathbf{x} = \sum_{\mathbf{w} \in \mathcal{W}} \mathbf{x}^{\mathbf{w}}, \, \mathbf{N}\mathbf{x}^{\mathbf{w}} = \mathbf{d}^{\mathbf{w}}, \, \forall \mathbf{w} \in \mathcal{W}\Big\}, \tag{1}$$

where $\mathbf{x}^{\mathbf{w}}$ is the flow vector attributed to OD pair $\mathbf{w}$.

In order to formulate appropriate optimization and inverse optimization problems, we next state the definition of the *Wardrop equilibrium*.

Definition 1 ([4])

A feasible flow $\mathbf{x}^* \in \mathcal{F}$ is a *Wardrop equilibrium* if for every OD pair $\mathbf{w} = (w_s, w_t) \in \mathcal{W}$, and any route connecting (w_s, w_t) with positive flow in $\mathbf{x}^*$, the cost of traveling along that route is no greater than the cost of traveling along any other route that connects (w_s, w_t).

In the above definition, the cost of traveling along a route is the sum of the costs of each of its constituent links.

2.1.2. Multi-class transportation network model

Denote by $|\tilde{\mathcal{U}}|$ the number of user (vehicle) classes. Let the original network be $\big(\tilde{\mathcal{V}}, \tilde{\mathcal{A}}, \tilde{\mathcal{W}}\big)$, where $\tilde{\mathcal{V}} = \big\{v_i : i \in [\![\tilde{\mathcal{V}}]\!]\big\}$, $\tilde{\mathcal{A}} = \big\{a_i : i \in [\![\tilde{\mathcal{A}}]\!]\big\}$, and $\tilde{\mathcal{W}} = \big\{\mathbf{w}_i : \mathbf{w}_i \overset{\text{def}}{=} (w_{si}, w_{ti}), i \in [\![\tilde{\mathcal{W}}]\!]\big\}$. We borrow the idea of making $|\tilde{\mathcal{U}}|$ copies of $\big(\tilde{\mathcal{V}}, \tilde{\mathcal{A}}, \tilde{\mathcal{W}}\big)$, each corresponding to a single vehicle class, to obtain an enlarged single-class network [2]. In particular, we construct a single-class network $(\mathcal{V}, \mathcal{A}, \mathcal{W})$, where

$$\mathcal{V} = \big\{v(i,u) : i \in [\![\tilde{\mathcal{V}}]\!], u \in [\![\tilde{\mathcal{U}}]\!]\big\}, \tag{2}$$

$$\mathcal{A} = \big\{a(i,u) : i \in [\![\tilde{\mathcal{A}}]\!], u \in [\![\tilde{\mathcal{U}}]\!]\big\}, \tag{3}$$

$$\mathcal{W} = \big\{\mathbf{w}(i,u) : \mathbf{w}(i,u) \overset{\text{def}}{=} (w_s(i,u), w_t(i,u)), i \in [\![\tilde{\mathcal{W}}]\!],\ u \in [\![\tilde{\mathcal{U}}]\!]\big\}. \tag{4}$$

We then can derive the node-link incidence matrix $\mathbf{N} \in \{0, 1, -1\}^{|\mathcal{V}| \times |\mathcal{A}|}$. Note that $|\mathcal{V}| = |\tilde{\mathcal{U}}||\tilde{\mathcal{V}}|$, and $|\mathcal{A}| = |\tilde{\mathcal{U}}||\tilde{\mathcal{A}}|$. For $i \in [\![\tilde{\mathcal{A}}]\!]$, $u \in [\![\tilde{\mathcal{U}}]\!]$, let $\mathbf{e}_{iu}$ denote the $|\mathcal{A}|$-dimensional vector with an entry equal to 1 corresponding to link $a(i,u)$ and all the other entries equal to 0. Also, for $i \in [\![\tilde{\mathcal{W}}]\!]$, $u \in [\![\tilde{\mathcal{U}}]\!]$, we denote by $d^{\mathbf{w}(i,u)}$ the flow demand of vehicle class u from node $w_s(i,u)$ to node $w_t(i,u)$. Then, we define $\mathbf{d}^{\mathbf{w}} \in \mathbb{R}^{|\mathcal{V}|}$ as for the single-class case. Accordingly, the set of feasible flow vectors $\mathcal{F}$ can be defined as in (1). For convenience, we vectorize a given demand matrix as $\mathbf{g} = \big(g_{iu};\, i \in [\![\tilde{\mathcal{W}}]\!], u \in [\![\tilde{\mathcal{U}}]\!]\big)$, where for $i \in [\![\tilde{\mathcal{W}}]\!]$, $u \in [\![\tilde{\mathcal{U}}]\!]$, g_{iu} denotes the flow demand of vehicle class u for OD pair i. In addition, we denote by $\mathbf{g}_u = \big(g_{iu};\, i \in [\![\tilde{\mathcal{W}}]\!]\big)$ the demand vector for vehicle class $u \in [\![\tilde{\mathcal{U}}]\!]$.

Write a feasible flow vector $\mathbf{x} \in \mathcal{F}$ as $\mathbf{x} = \big(x_{iu};\, i \in [\![\tilde{\mathcal{A}}]\!], u \in [\![\tilde{\mathcal{U}}]\!]\big)$, where x_{iu} denotes the flow on link $a(i,u)$. Let $\mathbf{x}_u = \big(x_{iu};\, i \in [\![\tilde{\mathcal{A}}]\!]\big)$ be the flow vector for vehicle class $u \in [\![\tilde{\mathcal{U}}]\!]$ and $\mathbf{x}_{a_i} = \big(x_{iu};\, u \in [\![\tilde{\mathcal{U}}]\!]\big)$ the flow vector of all vehicle classes corresponding to the ith *physical* link. We consider the following cost function:

$$\mathbf{t}(\mathbf{x}) = \big(t_{iu}(\mathbf{x}_{a_i});\, i \in [\![\tilde{\mathcal{A}}]\!], u \in [\![\tilde{\mathcal{U}}]\!]\big), \tag{5}$$

where the cost on a *physical* link does not depend on the flows elsewhere, but a *physical* link maps to $|\tilde{\mathcal{U}}|$ *conceptual* links, each of which corresponds to a vehicle class.

It is seen that the single-class model is actually a special case of the multi-class model, and that, formally, the multi-class model can be treated as an enlarged single-class model. Thus, in the following, we only need to consider general multi-class models. However, as is well known, due to coupling of flows from different classes of vehicles, some of the properties could be very different for a model with $|\tilde{\mathcal{U}}| > 1$ compared to the single-class case.

2.2. Forward problem and inverse problem IP-1

In this section we formulate the forward problem TAP and the inverse problem IP-1 for the multi-class transportation network $(\mathcal{V}, \mathcal{A}, \mathcal{W})$ defined in Sec. 2.1.2 (see (2)-(4)).

2.2.1. The forward problem

As in [8], here we refer to the Traffic Assignment Problem (TAP) as the *forward problem*, whose goal is to find the Wardrop equilibrium for a given transportation network $(\mathcal{V}, \mathcal{A}, \mathcal{W})$ with a given travel latency cost function $\mathbf{t}$ and a given OD demand vector $\mathbf{g}$. It is a well-known result that TAP can be formulated as a Variational Inequality (VI) problem $\mathrm{VI}(\mathbf{t}, \mathcal{F})$, defined as follows.

Definition 2

([7]). The VI problem, denoted as $\mathrm{VI}(\mathbf{t}, \mathcal{F})$, is to find an $\mathbf{x}^* \in \mathcal{F}$ s.t.

$$\mathbf{t}(\mathbf{x}^*)' (\mathbf{x} - \mathbf{x}^*) \geq 0, \quad \forall \mathbf{x} \in \mathcal{F}. \tag{6}$$

To proceed, let us first present a definition regarding the monotonicity of a cost function.

Definition 3

([4]). $\mathbf{t}(\cdot)$ is *strongly monotone* on $\mathcal{F}$ if there exists a constant $\eta > 0$ such that

$$(\mathbf{t}(\mathbf{x}) - \mathbf{t}(\mathbf{y}))' (\mathbf{x} - \mathbf{y}) \geq \eta \|\mathbf{x} - \mathbf{y}\|^2, \quad \forall \mathbf{x}, \mathbf{y} \in \mathcal{F}.$$

To ensure the existence and uniqueness of the solution to $\mathrm{VI}(\mathbf{t}, \mathcal{F})$, we need the following assumption.

Assumption A

$\mathbf{t}(\cdot)$ is strongly monotone on $\mathcal{F}$ and continuously differentiable on $\mathbb{R}_+^{|\mathcal{A}|}$. $\mathcal{F}$ is nonempty and contains an interior point (Slater's condition [19]).

Relating the Wardrop equilibrium to the VI problem, the following result is well-known in the literature.

Theorem 2.1 *([4]). Under Assump. A , a Wardrop equilibrium of the multi-class transportation network is a solution to VI($\mathbf{t}, \mathcal{F}$), where $\mathbf{t}, \mathcal{F}$ are given by* (5) *and* (1), *respectively.*

Remark 1

As noted in [18], Assump. A cannot be easily verified for general multi-class transportation networks. We therefore do not have any guarantee of always obtaining unique link flows for each and every class of vehicles.

2.2.2. BPR-type cost functions

To simplify the analysis, we now further specify the cost functions in (5). For each $i \in [\![\tilde{\mathcal{A}}]\!], u \in [\![\tilde{\mathcal{U}}]\!]$, we define the following generalized Bureau of Public Roads (BPR)-type travel time (latency cost) function [6, 18, 7]:

$$t_{iu}(\mathbf{x}) = t_{iu}^0 f\Big(\frac{\boldsymbol{\theta}'\mathbf{x}_{a_i}}{m_i}\Big), \tag{7}$$

where t_{iu}^0 is called the *free-flow travel time* for vehicle class u on link a_i (the ith *physical* link), $f(\cdot)$ is a "cornerstone" cost function, which satisfies $f(0) = 1$ and is strictly increasing and continuously differentiable on $\mathbb{R}_+$, m_i is the *effective flow capacity* of link a_i, and $\boldsymbol{\theta} = \big(\theta_u;\, u \in [\![\tilde{\mathcal{U}}]\!]\big)$ is a weight vector such that $\theta_u \geq 1$, $\forall u \in [\![\tilde{\mathcal{U}}]\!]$. As a special case, the single-class network corresponds to $|\tilde{\mathcal{U}}| = 1$ and $\theta_1 = 1$.

2.2.3. The inverse problem IP-1

In order to solve the forward problem, thus obtaining the Wardrop equilibrium flow vector $\mathbf{x}$, we need to know the cost function $\mathbf{t}$ and the OD demand vector $\mathbf{g}$. Now, assuming that we know the OD demand vector $\mathbf{g}$ and have observed the Wardrop equilibrium flow vector $\mathbf{x}$, we seek to formulate the inverse problem IP-1 (as an inverse VI problem, in particular), so as to estimate the travel latency cost function $\mathbf{t}$ (specifically, $f(\cdot)$ in (7)). To provide some insight, given $|\mathcal{K}|$ samples of the link flow vector $\mathbf{x}$, one can think of them as flow observations on $|\mathcal{K}|$ different networks (or subnetworks) which are nevertheless produced by the exact same cost function. The inverse problem formulation seeks to determine the cost function so that each flow observation is as close to an equilibrium as possible.

To that end, for a given $\epsilon > 0$, we define an ϵ-*approximate solution* to $\mathrm{VI}(\mathbf{t}, \mathcal{F})$ by changing the right-hand side of (6) to $-\epsilon$.

Definition 4

([7]). Given $\epsilon > 0$, $\hat{\mathbf{x}} \in \mathcal{F}$ is called an ϵ-*approximate solution* to $\mathrm{VI}(\mathbf{t}, \mathcal{F})$ if

$$\mathbf{t}(\hat{\mathbf{x}})'(\mathbf{x} - \hat{\mathbf{x}}) \geq -\epsilon, \quad \forall \mathbf{x} \in \mathcal{F}. \tag{8}$$

Assume now we are given $|\mathcal{K}|$ networks $(\mathcal{V}^{(k)}, \mathcal{A}^{(k)}, \mathcal{W}^{(k)})$, $k \in [\![\mathcal{K}]\!]$ (as a special case, these could be $|\mathcal{K}|$ replicas of the same network $(\mathcal{V}, \mathcal{A}, \mathcal{W})$), and the observed link flow data $\big(\mathbf{x}_{a_i}^{(k)} = \big(x_{iu}^{(k)};\, u \in [\![\tilde{\mathcal{U}}]\!]\big);\, i \in [\![\tilde{\mathcal{A}}^{(k)}]\!], k \in [\![\mathcal{K}]\!]\big)$. The inverse VI problem amounts to finding a function $\mathbf{t}$ such that $\mathbf{x}^{(k)}$ is an ϵ_k-approximate solution to $\mathrm{VI}(\mathbf{t}, \mathcal{F}^{(k)})$ for each k. Denoting $\boldsymbol{\epsilon} \stackrel{\text{def}}{=} (\epsilon_k;\, k \in [\![\mathcal{K}]\!])$, we can formulate the inverse VI problem as in [7]

$$\begin{aligned}
\min_{\mathbf{t}, \boldsymbol{\epsilon}} \quad & \|\boldsymbol{\epsilon}\| && (9)\\
\text{s.t.} \quad & \mathbf{t}(\mathbf{x}^{(k)})'(\mathbf{x} - \mathbf{x}^{(k)}) \geq -\epsilon_k, \quad \forall \mathbf{x} \in \mathcal{F}^{(k)}, k \in [\![\mathcal{K}]\!], \\
& \epsilon_k > 0, \quad \forall k \in [\![\mathcal{K}]\!],
\end{aligned}$$

where the optimization is over $\boldsymbol{\epsilon}$ and the selection of function $\mathbf{t}(\cdot)$ (that is, $f(\cdot)$).

Aiming at recovering a cost function that has both good data reconciling and generalization properties, we apply an estimation approach which expresses the function $f(\cdot)$ (recall (7)) in a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$ [7, 20]. In particular, by [7, Thm. 2], being a variant of [9, (6)], the inverse VI problem (9) can be reformulated as a Quadratic Program (QP):

(invVI-1)

$$\min_{\mathbf{y},\boldsymbol{\epsilon}} \ \|\boldsymbol{\epsilon}\| + \gamma \|f\|_{\mathcal{H}}^2 \tag{10}$$

$$\text{s.t. } \mathbf{e}_{iu}' \mathbf{N}_k' \mathbf{y}^{\mathbf{w}} \le t_{iu}^0 f\left(\frac{\boldsymbol{\theta}'\mathbf{x}_{a_i}^{(k)}}{m_i^{(k)}}\right), \qquad \forall i \in [\![\tilde{\mathcal{A}}^{(k)}]\!],\ u \in [\![\tilde{\mathcal{U}}]\!],\ \mathbf{w} \in \mathcal{W}^{(k)},\ k \in [\![\mathcal{K}]\!], \tag{11}$$

$$\sum_{i=1}^{|\tilde{\mathcal{A}}^{(k)}|} \left(\sum_{u=1}^{|\tilde{\mathcal{U}}|} t_{iu}^0 x_{iu}\right) f\left(\frac{\boldsymbol{\theta}'\mathbf{x}_{a_i}^{(k)}}{m_i^{(k)}}\right) - \sum_{\mathbf{w}\in\mathcal{W}_k} \left(\mathbf{d}^{\mathbf{w}}\right)' \mathbf{y}^{\mathbf{w}} \le \epsilon_k, \qquad \forall k \in [\![\mathcal{K}]\!], \tag{12}$$

$$f\left(\frac{\boldsymbol{\theta}'\mathbf{x}_{a_i}^{(k)}}{m_i^{(k)}}\right) < f\left(\frac{\boldsymbol{\theta}'\mathbf{x}_{a_{\tilde{i}}}^{(k)}}{m_{\tilde{i}}^{(k)}}\right), \qquad \forall i, \tilde{i} \in [\![\tilde{\mathcal{A}}^{(k)}]\!] \ \text{ s.t. } \frac{\boldsymbol{\theta}'\mathbf{x}_{a_i}^{(k)}}{m_i^{(k)}} < \frac{\boldsymbol{\theta}'\mathbf{x}_{a_{\tilde{i}}}^{(k)}}{m_{\tilde{i}}^{(k)}};\ \forall k \in [\![\mathcal{K}]\!], \tag{13}$$

$$\boldsymbol{\epsilon} \ge \mathbf{0}, \quad f \in \mathcal{H},$$

$$f(0) = 1, \tag{14}$$

where $\mathbf{N}_k$ is the node-link incidence matrix of the k-th network, $\mathbf{y} = \left(\mathbf{y}^{\mathbf{w}} \in \mathbb{R}^{|\mathcal{V}^{(k)}|};\ \mathbf{w} \in \mathcal{W}^{(k)}, k \in [\![\mathcal{K}]\!]\right)$, $\boldsymbol{\epsilon} = (\epsilon_k;\ k \in [\![\mathcal{K}]\!])$ are decision vectors, $\mathbf{y}^{\mathbf{w}}$ is a dual variable which can be interpreted as the "price" of $\mathbf{d}^{\mathbf{w}}$, $\gamma > 0$ is a regularization parameter, $\|f\|_{\mathcal{H}}^2$ denotes the squared norm of $f(\cdot)$ in $\mathcal{H}$, (11) is for dual feasibility, (12) is the suboptimality (primal-dual gap) constraint, (13) enforces $f(\cdot)$ to be increasing, and (14) is for normalization purposes (see (7)). Note that a smaller γ should result in recovering a "tighter" $f(\cdot)$ in terms of data reconciliation whereas a bigger γ would lead to a "better" $f(\cdot)$ in terms of generalization properties.

It can be seen that the above formulation is still too abstract for us to solve, because it is an optimization over functions. To make it tractable, in the following, we specify $\mathcal{H}$ by picking its *reproducing kernel* [20] as a polynomial $\phi(x, y) \stackrel{\text{def}}{=} (c + xy)^n$ for some choice of $c \ge 0$ and $n \in \mathbb{N}$. Then, writing

$$\phi(x, y) = (c + xy)^n = \sum_{j=0}^{n} \binom{n}{j} c^{n-j} x^j y^j,$$

by [20, (3.2), (3.3), and (3.6)], we instantiate invVI-1 as [10]

(invVI-2)

$$\min_{\boldsymbol{\beta},\mathbf{y},\boldsymbol{\epsilon}} \ \|\boldsymbol{\epsilon}\| + \gamma \sum_{j=0}^{n} \frac{\beta_j^2}{\binom{n}{j} c^{n-j}} \tag{15}$$

$$\text{s.t. } \mathbf{e}_{iu}' \mathbf{N}_k' \mathbf{y}^{\mathbf{w}} \le t_{iu}^0 \sum_{j=0}^{n} \beta_j \left(\frac{\boldsymbol{\theta}'\mathbf{x}_{a_i}^{(k)}}{m_i^{(k)}}\right)^j, \qquad \forall i \in [\![\tilde{\mathcal{A}}^{(k)}]\!],\ u \in [\![\tilde{\mathcal{U}}]\!],\ \mathbf{w} \in \mathcal{W}^{(k)},\ k \in [\![\mathcal{K}]\!],$$

$$\sum_{i=1}^{|\tilde{\mathcal{A}}^{(k)}|}\left(\sum_{j=0}^{n}\beta_j\left(\frac{\boldsymbol{\theta}'\mathbf{x}_{a_i}^{(k)}}{m_i^{(k)}}\right)^j\right)\sum_{u=1}^{|\tilde{\mathcal{U}}|}t_{iu}^0 x_{iu}-\sum_{\mathbf{w}\in\mathcal{W}_k}\left(\mathbf{d^w}\right)'\mathbf{y^w}\le\epsilon_k,\qquad\forall k\in[\![\mathcal{K}]\!],$$

$$\sum_{j=0}^{n}\beta_j\left(\frac{\boldsymbol{\theta}'\mathbf{x}_{a_i}^{(k)}}{m_i^{(k)}}\right)^j<\sum_{j=0}^{n}\beta_j\left(\frac{\boldsymbol{\theta}'\mathbf{x}_{a_{\tilde{i}}}^{(k)}}{m_{\tilde{i}}^{(k)}}\right)^j,\qquad\forall i,\,\tilde{i}\in[\![\tilde{\mathcal{A}}^{(k)}]\!]\ \text{ s.t. }\frac{\boldsymbol{\theta}'\mathbf{x}_{a_i}^{(k)}}{m_i^{(k)}}<\frac{\boldsymbol{\theta}'\mathbf{x}_{a_{\tilde{i}}}^{(k)}}{m_{\tilde{i}}^{(k)}};\,\forall k\in[\![\mathcal{K}]\!],$$

$$\boldsymbol{\epsilon}\ge\mathbf{0},$$

$$\beta_0=1,$$

where the function $f(\cdot)$ in invVI-1 is parametrized by $\boldsymbol{\beta}=(\beta_i;\ i=0,1,\ldots,n)$. Assuming an optimal $\boldsymbol{\beta}^*=\left(\beta_j^*;\ j=0,1,\ldots,n\right)$ is obtained by solving (15), then our estimator for the cost function $f(\cdot)$ is

$$\hat{f}\left(x\right)=\sum_{j=0}^{n}\beta_j^* x^j=1+\sum_{j=1}^{n}\beta_j^* x^j. \tag{16}$$

It is worth pointing out that, in the above QP formulations, the parameter vector $\boldsymbol{\theta}$ and the set of vehicle classes $\tilde{\mathcal{U}}$ have been assumed to be the same for all $|\mathcal{K}|$ networks. We note that in (7) what essentially gets involved is only the weighed sum of link flows from different classes of vehicles (other than the link flow of each single vehicle class). It turns out that, similar to our previous work on single-class networks [8, 9], we are still very likely to be able to recover the cost functions with satisfactory accuracy from such weighted sum of link flows.

3. The Joint Problem

Different from previous works, in this section we solve a joint problem to recover the cost function $f(\cdot)$ (specifically, the coefficient vector $\boldsymbol{\beta}=(\beta_j;\ j=0,1,\ldots,n)$ with $\beta_0=1$; cf. (16)), hence $\mathbf{t}$, and adjust the OD demand vector $\mathbf{g}$. To simplify notation, for any given feasible $\boldsymbol{\beta}$ (determining $f(\cdot)$, hence $\mathbf{t}$) and $\mathbf{g}$, let $\mathbf{x}\left(\boldsymbol{\beta},\mathbf{g}\right)=\left(x_{iu}\left(\boldsymbol{\beta},\mathbf{g}\right);\, i\in[\![\tilde{\mathcal{A}}]\!],u\in[\![\tilde{\mathcal{U}}]\!]\right)$ be the optimal solution to VI($\mathbf{t},\mathcal{F}$).

Assume that we have observed an equilibrium flow vector, denoted by $\mathbf{x}^*=\left(x_{iu}^*;\ i\in[\![\tilde{\mathcal{A}}]\!],u\in[\![\tilde{\mathcal{U}}]\!]\right)$, and that an initial demand vector $\mathbf{g}^0$ is available. We seek to solve the following bi-level optimization problem, which is a variant of [11, (1)-(2)], [14, (1)-(8)], and [9, (9)]:

$$\text{(BiLev)}\quad \min\ F\left(\boldsymbol{\beta},\mathbf{g}\right)\overset{\text{def}}{=}\gamma_1\sum_{i=1}^{|\tilde{\mathcal{W}}|}\sum_{u=1}^{|\tilde{\mathcal{U}}|}\left(g_{iu}-g_{iu}^0\right)^2+\gamma_2\sum_{\check{i}=1}^{|\tilde{\mathcal{A}}|}\sum_{\check{u}=1}^{|\tilde{\mathcal{U}}|}\left(x_{\check{i}\check{u}}\left(\boldsymbol{\beta},\mathbf{g}\right)-x_{\check{i}\check{u}}^*\right)^2 \tag{17}$$

$$\text{s.t.}\quad \boldsymbol{\beta}\ge\mathbf{0},\,\beta_0=1,\,\mathbf{g}\ge\mathbf{0},$$

where $\gamma_1,\gamma_2\ge0$ are two weight parameters; the first term penalizes moving too far away from the initial demand, and the second term ensures that the optimal solution to TAP is close to the flow observation. Note that BiLev (17) is more general than its counterparts (IP-2, in particular) considered in [11, 14, 9].

It is also worth pointing out that $F(\boldsymbol{\beta}, \mathbf{g})$ is bounded below by 0 which guarantees the convergence of the algorithm (see Alg. 1) that we will apply.

To solve BiLev numerically, thus alternatively recovering $\boldsymbol{\beta}$ and adjusting $\mathbf{g}$ in an iterative manner, we leverage a gradient-based algorithm (Alg. 1). Let us first derive an approximation to the gradient of $F(\boldsymbol{\beta}, \mathbf{g})$ (defined in (17)) with respect to $\mathbf{g}$. To that end, fix $\check{i} \in [\![\tilde{\mathcal{A}}]\!]$, $\check{u} \in [\![\tilde{\mathcal{U}}]\!]$, and a feasible pair $(\boldsymbol{\beta}, \mathbf{g})$. By flow conservation, we have

$$x_{\check{i}\check{u}}(\boldsymbol{\beta}, \mathbf{g}) = \sum_{i \in [\![\tilde{\mathcal{W}}]\!]} \sum_{r \in \mathcal{R}_{\check{u}}^{i}} \delta_r^{a(\check{i},\check{u})} p_{\check{u}}^{ir} g_{i\check{u}} = \sum_{i \in [\![\tilde{\mathcal{W}}]\!]} g_{i\check{u}} \sum_{r \in \mathcal{R}_{\check{u}}^{i}} \delta_r^{a(\check{i},\check{u})} p_{\check{u}}^{ir}, \tag{18}$$

where $\mathcal{R}_{\check{u}}^{i}$ denotes the set of feasible routes associated with OD pair i and user class $\check{u}$, $p_{\check{u}}^{ir}$ is the probability for user class $\check{u}$ to select route $r \in \mathcal{R}_{\check{u}}^{i}$, and

$$\delta_r^{a(\check{i},\check{u})} \stackrel{\text{def}}{=} \begin{cases} 1, \text{ if route } r \text{ uses link } a(\check{i}, \check{u}), \\ 0, \text{ otherwise.} \end{cases} \tag{19}$$

Assuming that the route probabilities are locally constant, then (18) implies [14, 11]

$$\frac{\partial x_{\check{i}\check{u}}(\boldsymbol{\beta}, \mathbf{g})}{\partial g_{iu}} = \begin{cases} \sum_{r \in \mathcal{R}_{\check{u}}^{i}} \delta_r^{a(\check{i},\check{u})} p_{\check{u}}^{ir}, \text{ if } \check{u} = u, \\ 0, \text{ otherwise,} \end{cases} \qquad \forall i \in [\![\tilde{\mathcal{W}}]\!],\ u \in [\![\tilde{\mathcal{U}}]\!]. \tag{20}$$

Further, for OD pair $i \in [\![\tilde{\mathcal{W}}]\!]$ and vehicle class $u \in [\![\tilde{\mathcal{U}}]\!]$, considering only the shortest route $r_{iu}(\boldsymbol{\beta}, \mathbf{g})$ based on the travel latency cost (i.e., travel time), we have

$$\begin{aligned} \frac{\partial x_{\check{i}\check{u}}(\boldsymbol{\beta}, \mathbf{g})}{\partial g_{iu}} \approx \delta_{r_{iu}(\boldsymbol{\beta}, \mathbf{g})}^{a(\check{i},\check{u})} &= \begin{cases} 1, \text{ if } a(\check{i}, \check{u}) \in r_{iu}(\boldsymbol{\beta}, \mathbf{g}), \\ 0, \text{ otherwise,} \end{cases} \\ &= \begin{cases} 1, \text{ if } \check{u} = u \text{ and } a(\check{i}, \check{u}) \in r_{iu}(\boldsymbol{\beta}, \mathbf{g}), \\ 0, \text{ otherwise,} \end{cases} \end{aligned} \tag{21}$$

where $a(\check{i}, \check{u}) \in r_{iu}(\boldsymbol{\beta}, \mathbf{g})$ indicates that the route $r_{iu}(\boldsymbol{\beta}, \mathbf{g})$ uses link $a(\check{i}, \check{u})$. Note that $a(\check{i}, \check{u}) \notin r_{iu}(\boldsymbol{\beta}, \mathbf{g})$ for all $\check{u} \neq u$. Note also that we have assumed the partial derivatives do exist; if not, one can replace them with subgradients. Such partial derivatives as in the BiLev setting typically do not have an exact analytical expression and we in turn use a certain approximation technique; a comprehensive discussion on the existence and approximation of such type of partial derivatives can be found in [21]. By (21) we obtain an approximation to the Jacobian matrix

$$\left[\frac{\partial x_{\check{i}\check{u}}(\boldsymbol{\beta}, \mathbf{g})}{\partial g_{iu}};\ \check{i} \in [\![\tilde{\mathcal{A}}]\!],\ \check{u} \in [\![\tilde{\mathcal{U}}]\!],\ i \in [\![\tilde{\mathcal{W}}]\!],\ u \in [\![\tilde{\mathcal{U}}]\!]\right]. \tag{22}$$

Let us now compute the gradient of $F(\boldsymbol{\beta}, \mathbf{g})$ with respect to $\mathbf{g}$. We have

$$\nabla_{\mathbf{g}} F(\boldsymbol{\beta}, \mathbf{g}) = \left(\frac{\partial F(\boldsymbol{\beta}, \mathbf{g})}{\partial g_{iu}};\ i \in [\![\tilde{\mathcal{W}}]\!],\ u \in [\![\tilde{\mathcal{U}}]\!]\right)$$

$$= \left(2\gamma_1 \left(g_{iu} - g^0_{iu} \right) + 2\gamma_2 \sum_{\check{i}=1}^{|\tilde{\mathcal{A}}|} \sum_{\check{u}=1}^{|\tilde{\mathcal{U}}|} \left(x_{\check{i}\check{u}} \left(\boldsymbol{\beta}, \mathbf{g} \right) - x^*_{\check{i}\check{u}} \right) \frac{\partial x_{\check{i}\check{u}} \left(\boldsymbol{\beta}, \mathbf{g} \right)}{\partial g_{iu}}; \, i \in [\![\tilde{\mathcal{W}}]\!], \, u \in [\![\tilde{\mathcal{U}}]\!] \right)$$

$$= \left(2\gamma_1 \left(g_{iu} - g^0_{iu} \right) + 2\gamma_2 \sum_{\check{i}=1}^{|\tilde{\mathcal{A}}|} \left(x_{\check{i}u} \left(\boldsymbol{\beta}, \mathbf{g} \right) - x^*_{\check{i}u} \right) \frac{\partial x_{\check{i}u} \left(\boldsymbol{\beta}, \mathbf{g} \right)}{\partial g_{iu}}; \, i \in [\![\tilde{\mathcal{W}}]\!], \, u \in [\![\tilde{\mathcal{U}}]\!] \right). \tag{23}$$

Remark 2

Similar to [9, 13], the reasons why we consider only the shortest routes for the purpose of calculating the Jacobian include: (i) GPS navigation is widely-used by vehicle drivers so that they tend to always select the shortest routes between their OD pairs. (ii) Considering the shortest routes only would significantly simplify the calculation of the route-choice probabilities. (iii) Extensive numerical experiments show that such an approximation as (21) for the Jacobian matrix performs satisfactorily well.

We summarize the procedures for alternatively recovering the cost function and adjusting the OD demand vector as Alg. 1, whose convergence will be proven in the following proposition.

Proposition 3.1 *Alg. 1 converges.*

Proof: If the initial demand vector $\mathbf{g}^0$ satisfies $F\left(\boldsymbol{\beta}^0, \mathbf{g}^0\right) = 0$, then, by Step 0, the algorithm terminates (trivial case). Otherwise, we have $F\left(\boldsymbol{\beta}^0, \mathbf{g}^0\right) > 0$, and it is seen from (17) that the objective function $F\left(\boldsymbol{\beta}, \mathbf{g}\right)$ has a lower bound 0. In addition, by the line search and the update steps (Steps 3.2 and 4.1, in particular), we obtain

$$F\left(\boldsymbol{\beta}^l, \mathbf{g}^{l+1}\right) = F\left(\boldsymbol{\beta}^l, \mathbf{g}^l + \alpha^l \bar{\mathbf{h}}^l\right) = \min_{\alpha \in \mathcal{S}} F\left(\boldsymbol{\beta}^l, \mathbf{g}^l + \alpha \bar{\mathbf{h}}^l\right) \leq F\left(\boldsymbol{\beta}^l, \mathbf{g}^l\right), \, \forall l, \tag{24}$$

where the last inequality holds due to $0 \in \mathcal{S}$. In addition, by Step 4.2 it is assured that $F\left(\boldsymbol{\beta}^{l+1}, \mathbf{g}^{l+1}\right) \leq F\left(\boldsymbol{\beta}^l, \mathbf{g}^{l+1}\right)$ (cf. Remark 3), which, combined with (24), indicates that the nonnegative objective function $F(\boldsymbol{\beta}, \mathbf{g})$ in (17) is non-increasing. Thus, by the well-known monotone convergence theorem, the convergence of the algorithm can be guaranteed. $\square$

Remark 3

Regarding Steps 0 and 4.2 of Alg. 1, a natural question could be raised: Shouldn't $\mathbf{x}^l$ ($\forall l \geq 0$) obtained in that way be equal to $\mathbf{x}^*$? We note that $\mathbf{x}^l$ is typically not equal to $\mathbf{x}^*$, because $\mathbf{x}^*$ does not necessarily satisfy flow conservations for all OD pairs corresponding to $\mathbf{g}^l$ when $\mathbf{g}^l \neq \mathbf{g}^*$ ($\mathbf{g}^*$ denotes the ground-truth OD demand vector), but $\mathbf{x}^l$ does. To put it another way, if $\mathbf{g}^l \neq \mathbf{g}^*$, then we have $\mathbf{x}^* \notin \mathcal{F}^l$, but $\mathbf{x}^l \in \mathcal{F}^l$, where $\mathcal{F}^l$ is the feasible set $\mathcal{F}$ (see (1)) instantiated under demand vector $\mathbf{g}^l$. It is also worth pointing out that, mathematically, such schemes for "tuning" the cost function as elaborated in Steps 0 and 4.2, would serve to help decrease the objective function value such that $F(\boldsymbol{\beta}^{l+1}, \mathbf{g}^{l+1}) \leq F(\boldsymbol{\beta}^l, \mathbf{g}^{l+1})$.

Algorithm 1 Alternatively recovering the travel latency cost function and adjusting the OD demand vector.

Input: $(\mathcal{V}, \mathcal{A}, \mathcal{W})$: road network; $\mathbf{x}^*$: the observed flow vector; $\mathbf{g}^0$: the initial demand vector; ρ, T: two positive integer parameters; $\epsilon_1 \geq 0$, $\epsilon_2 > 0$: two real parameters.

Step 0: Initialization. Using $\mathbf{g}^0$ and $\mathbf{x}^*$ as the input, solve (15) to obtain $\boldsymbol{\beta}^0$. Plug $\boldsymbol{\beta}^0$ into (16) to obtain an initial cost function $\hat{f}^0(\cdot)$; then, with this cost function and the demand vector $\mathbf{g}^0$, solve the forward problem VI$(\mathbf{t}, \mathcal{F})$ (using the Method of Successive Averages (MSA) [18]; see Alg. 2) to obtain $\mathbf{x}^0$. Set $l = 0$. If $F\left(\boldsymbol{\beta}^0, \mathbf{g}^0\right) = 0$, terminate and return $\boldsymbol{\beta}^0$, $\mathbf{g}^0$; otherwise, go onto Step 1.

Step 1: Computation of a descent direction. Calculate $\mathbf{h}^l = -\nabla_{\mathbf{g}} F\left(\boldsymbol{\beta}^l, \mathbf{g}^l\right)$ by (23).

Step 2: Calculation of a search direction. For $i \in [\![\mathcal{W}]\!]$, $u \in [\![\mathcal{U}]\!]$, set

$$\bar{h}_{iu}^l = \begin{cases} h_{iu}^l, \text{ if } \left(g_{iu}^l > \epsilon_1\right) \text{ or } \left(g_{iu}^l \leq \epsilon_1 \text{ and } h_{iu}^l > 0\right), \\ 0, \text{ otherwise.} \end{cases}$$

Step 3: Determination of a step-size using Armijo-type line search.

3.1: Calculate the maximum possible step-size $\alpha_{\max}^l = \min\left\{-g_{iu}^l / \bar{h}_{iu}^l;\ \bar{h}_{iu}^l < 0,\ i \in [\![\mathcal{W}]\!],\ u \in [\![\mathcal{U}]\!]\right\}$.

3.2: Determine $\alpha^l = \arg\min_{\alpha \in \mathcal{S}} F\left(\boldsymbol{\beta}^l, \mathbf{g}^l + \alpha \bar{\mathbf{h}}^l\right)$ with $\mathcal{S} \overset{\text{def}}{=} \left\{\alpha_{\max}^l, \alpha_{\max}^l/\rho, \alpha_{\max}^l/\rho^2, \ldots, \alpha_{\max}^l/\rho^T, 0\right\}$.

Step 4: Update and termination.

4.1: Set $\mathbf{g}^{l+1} = \mathbf{g}^l + \alpha^l \bar{\mathbf{h}}^l$. If $\frac{F\left(\boldsymbol{\beta}^l, \mathbf{g}^l\right) - F\left(\boldsymbol{\beta}^l, \mathbf{g}^{l+1}\right)}{F\left(\boldsymbol{\beta}^0, \mathbf{g}^0\right)} < \epsilon_2$, stop iteration and return $\boldsymbol{\beta}^l$, $\mathbf{g}^{l+1}$; otherwise, go onto Step 4.2.

4.2: Using $\mathbf{g}^{l+1}$ and $\mathbf{x}^*$ as the input, solve (15) to obtain $\boldsymbol{\beta}^{l+1}$. Plug $\boldsymbol{\beta}^{l+1}$ into (16) to obtain a new cost function $\hat{f}^{l+1}(\cdot)$; then, with this cost function and the demand vector $\mathbf{g}^{l+1}$, solve VI$(\mathbf{t}, \mathcal{F})$ to obtain $\mathbf{x}^{l+1}$. If $F(\boldsymbol{\beta}^{l+1}, \mathbf{g}^{l+1}) > F(\boldsymbol{\beta}^l, \mathbf{g}^{l+1})$, reset $\boldsymbol{\beta}^{l+1} = \boldsymbol{\beta}^l$. Set $l = l + 1$ and return to Step 1.

Algorithm 2 Method of Successive Averages (MSA) [18]

Input: $(\tilde{\mathcal{V}}, \tilde{\mathcal{A}}, \tilde{\mathcal{W}})$: road network; $\tilde{\mathcal{U}}$: set of vehicle classes; $f(\cdot)$: cornerstone cost function in (7); $\mathbf{g}_u$, $u \in [\![\tilde{\mathcal{U}}]\!]$: demand vector for vehicle class u; $\epsilon_3 > 0$: a real parameter; L: maximum number of iterations.

1: **Step 0:** Initialization. Initialize link flows $x_{iu}^{\ell} = 0$ for $i \in [\![\tilde{\mathcal{A}}]\!], u \in [\![\tilde{\mathcal{U}}]\!]$; set iteration counter $\ell = 0$.

2: **Step 1:** Compute new extremal flows. Set $\ell = \ell + 1$.

1.1: Update link travel costs based on current link flows: $t_{iu}^{\ell} = t_{iu}(x_{i1}^{\ell-1}, \dots, x_{i|\tilde{\mathcal{U}}|}^{\ell-1})$, $\forall i \in [\![\tilde{\mathcal{A}}]\!], u \in [\![\tilde{\mathcal{U}}]\!]$.

1.2: Carry out "all-or-nothing" assignment of the demands $\mathbf{g}_u$ on current shortest paths to obtain y_{iu}^{ℓ}.

3: **Step 2:** Update link flows via

$$x_{iu}^{\ell} = x_{iu}^{\ell-1} + \lambda^{\ell} \left(y_{iu}^{\ell} - x_{iu}^{\ell-1} \right),$$

where $\lambda^{\ell} = 1/\ell$.

4: **Step 3:** Stopping criterion (slightly different than that in [18]). Compute the *Relative Gap* (*RG*)

$$\mathrm{RG} = \frac{\left\| \mathbf{x}^{\ell} - \mathbf{x}^{\ell-1} \right\|}{\|\mathbf{x}^{\ell}\|}.$$

If RG $< \epsilon_3$ or $\ell \geq L$, terminate; otherwise, return to Step 1.

Remark 4

Alg. 1 is a variant of the algorithms proposed in [11] and [12]. We use a different method to calculate the *step-sizes* (resp., *Jacobian matrix*) than that in [11] (resp., [12]). Moreover, as a subroutine, Alg. 2 is borrowed from [18], which has the advantages of being easy to implement and numerically stable when applied to multi-class models.

Remark 5

Let us fix $\gamma_2 = 1$ in (17) hereafter. Intuitively, the closer the initial $\mathbf{g}^0$ to the ground truth $\mathbf{g}^*$, the larger the γ_1 we should set; otherwise the contribution of the first term to the objective function will be small. In practice, however, we typically do not have exact information about how far $\mathbf{g}^0$ is from $\mathbf{g}^*$; we therefore need to appropriately tune γ_1. One possible criterion is that, fixing the parameters involved in Alg. 1, a "good" γ_1 should lead to a reduction of the objective function value of BiLev as much as possible. In Sec. 4 we will present our findings of how to set an appropriate γ_1 through numerical studies using a benchmark network.

Remark 6

Note that, formally, if $|\tilde{\mathcal{U}}| = 1$, then all the above analytical results derived for the multi-class transportation networks reduce to the single-class case. We should keep in mind, however, that the behavior of the two models could be quite different. For instance, Assump. A (the strong monotonicity assumption) on the cost function is hard to verify for general multi-class networks ($|\tilde{\mathcal{U}}| > 1$).

4. Numerical results

In this section, we elaborate on numerical experiments conducted on both benchmark networks [22] and an actual road network. First, over three benchmark networks ranging from small-sized to large-sized (Sioux-Falls, Berlin-Tiergarten, and Anaheim) we adopt the single-class model by assuming only one class of vehicles (private cars). Then, for each of the benchmark networks, we simulate a multi-class model by assuming we have both private cars, indexed by Class 1, and commercial trucks, indexed by Class 2, thus ending up with a multi-class model where $|\tilde{\mathcal{U}}| = 2$. Finally, leveraging actual traffic data and recently developed techniques, we apply our approach to a large highway network of Eastern Massachusetts (EMA), over which both a single-class (private cars only) model and a multi-class ($|\tilde{\mathcal{U}}| = 2$; both private cars and commercial trucks) model are considered.

In all the benchmark network single-class model scenarios, for each OD pair, we obtain the initial demand by scaling the ground truth demand (available via [22]) with a uniform distribution over $[0.9, 1.1]$. In all the benchmark network multi-class model scenarios, we divide the original ground truth demand (available via [22]) proportionally, 80% for private cars and 20% for commercial trucks, to obtain the ground truth demand vectors accordingly. For each OD pair and for each vehicle class, the initial demand is obtained by scaling the corresponding ground truth demand with a uniform distribution over $[0.9, 1.1]$. On the other hand, for the EMA highway network scenarios, we leverage real traffic data to estimate the initial demand vectors; for details, see the Appendix.

For both single-class and multi-class models, we take the ground truth $f(\cdot)$ to be $f(z) = 1+0.15z^4,\ z \geq 0$, for the Sioux-Falls and Anaheim networks, and $f(z) = 1 + z^4,\ z \geq 0$, for the Berlin-Tiergarten network. Whereas, for the EMA highway subnetwork we do not have a ground truth $f(\cdot)$; we take $f(z) = 1 + 0.15z^4,\ z \geq 0$, (the well-known BPR function) as a reference. Regardless of the benchmark networks or the EMA highway network, in all the multi-class model scenarios, taking account of the relatively lower speeds and larger sizes for trucks compared to cars, we assume the flow weight vector to be $\boldsymbol{\theta} = (1.0, 2.0)$, and assume $t^0_{i1} = 1.0 \times t^0_i$ and $t^0_{i2} = 1.1 \times t^0_i$, where t^0_i is the *reference free-flow travel time* (available via [22]) for any *physical* link indexed by i.

For each of the benchmark network, the observed flow vector $\mathbf{x}^*$ is generated by solving the forward problem $\mathrm{VI}(\mathbf{t}, \mathcal{F})$ via Alg. 2 (with the corresponding ground truth demand and cost function as input). While for the EMA highway network, $\mathbf{x}^*$ is inferred from the collected actual speed data and the flow capacity data; see the Appendix for details.

First, to investigate how the settings of γ_1 and γ_2 in (17) would affect the performance of Alg. 1 (in terms of the reduction of the objective function value of BiLev), we conduct numerical experiments over the Sioux-Falls benchmark network single-class model by fixing $\gamma_2 = 1$ and letting γ_1 vary over a candidate set $\{0, 0.001, 0.01, 0.1, 1, 10, 100, 1000\}$. We set $n = 6$, $c = 3.5$, and $\gamma = 1.0$ in (15). When implementing Algs. 1 and 2, we set $\rho = 2$, $T = 10$, $\epsilon_1 = 0$, $\epsilon_2 = 10^{-20}$, $\epsilon_3 = 10^{-6}$, and $L = 1000$. The key outputs are summarized in Tab. 1, from which we see that under the particular settings of this study, a relatively small γ_1 ($0 \leq \gamma_1 \leq 1$) would lead to a significant reduction of the objective function value of BiLev. We should keep in mind, however, that as noted in Remark 5, tuning γ_1 would be typically inevitable for a different scenario, and we could select a "good" γ_1 that would lead to a reduction of the objective function value of BiLev as much as possible.

Next, to ensure comparison fairness and without loss of generality, for all scenarios we universally take $\gamma_1 = 1$ and $\gamma_2 = 1$ in (17). When implementing Algs. 1 and 2, we set $\rho = 2$, $T = 10$, $\epsilon_1 = 0$, $\epsilon_2 = 10^{-20}$, $\epsilon_3 = 10^{-6}$, and $L = 1000$; these parameters could be tuned representing the trade-off between the computation burden and the output accuracy.

In the following, for each and every benchmark network we plot three types of key quantities at each iteration l of Alg. 1: the normalized objective function value $F^l/F^0 \overset{\text{def}}{=} F(\boldsymbol{\beta}^l, \mathbf{g}^l)/F(\boldsymbol{\beta}^0, \mathbf{g}^0)$, the normalized demand difference w.r.t. ground truth $\|\mathbf{g}^l - \mathbf{g}^*\|/\|\mathbf{g}^*\|$ for the single-class model (resp., $\|\mathbf{g}_u^l - \mathbf{g}_u^*\|/\|\mathbf{g}_u^*\|$ for the multi-class model), where $\mathbf{g}^*$ (resp., $\mathbf{g}_u^*$) denotes the ground truth, and the cost function estimates $\hat{f}^l(\cdot)$. On the other hand, for the EMA highway network, since we do not have the ground truth demand vector, we only plot the other two types of key quantities. Moreover, for convenience of comparisons, in Tabs. 2 and 3 we summarize some of the results in detail.

4.1. Sioux-Falls network

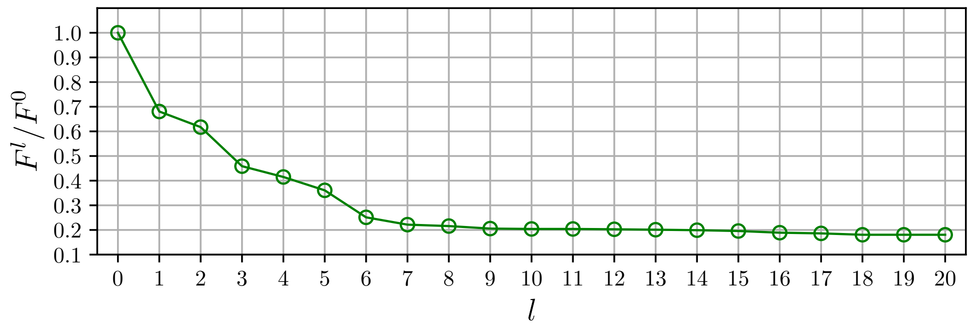
The Sioux-Falls network contains 24 zones (hence, $24 \times (24-1) = 552$ OD pairs), 24 nodes, and 76 links.

Single-class model

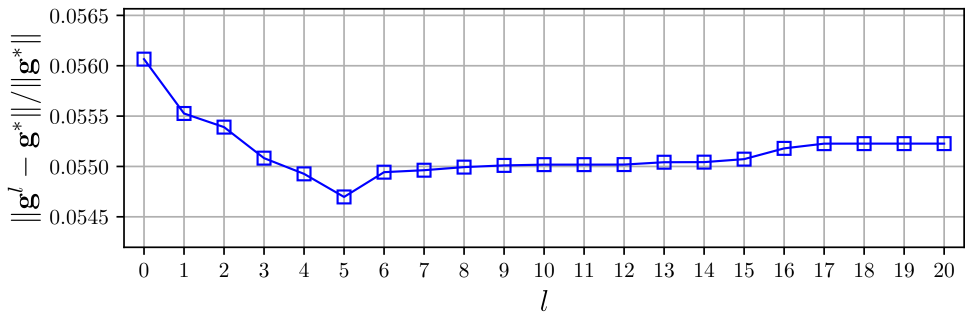
We use $n = 6$, $c = 3.5$, and $\gamma = 1.0$ in (15); note that, for a given network, using simulated data (demand vectors and the corresponding Wardrop equilibrium link flows), the values of n, c, γ can be determined by cross-validation and then applied to new data. We plot selected outputs after executing Alg. 1 in Fig. 1. Fig. 1a shows that, after 20 iterations, the objective function value of BiLev is reduced by more than 80%. Fig. 1b shows that, although the distance between the adjusted demand and the ground truth demand does not keep decreasing as the iteration count increases, the distance changes very slightly, meaning that the adjustment procedure does not alter the initial demand much. Fig. 1c shows that, a "rough" initial demand vector, combined with the observed flow vector, can already enable us to obtain a very accurate estimate for the cost function and, as the iteration count increases, the estimate of the cost function gets closer and closer to the ground truth (see the green curve corresponding to $l = 20$, which is the closest to the red ground truth).

Multi-class model

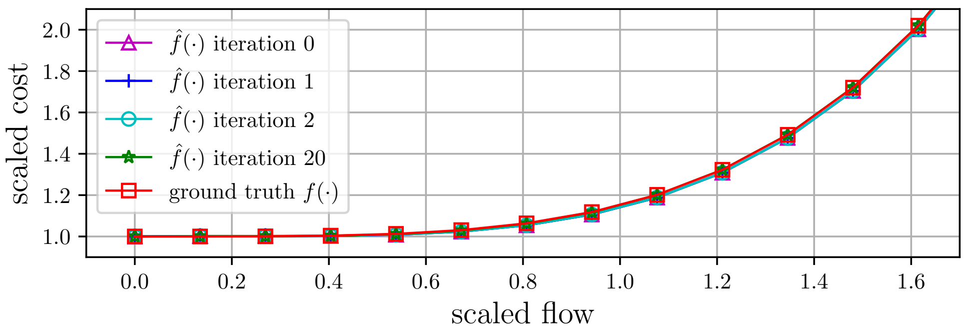
We use $n = 6$, $c = 3.5$, and $\gamma = 1.0$ in (15), and we plot the outputs in Fig. 2. Fig. 2a shows that, after 5 iterations, the objective function value of BiLev is reduced by about 35%. Fig. 2b shows that, for either vehicle class (private car or commercial truck), though the distance between the adjusted demand and the ground truth would not always keep decreasing as the iteration count increases, the distance changes very slightly, meaning that the adjustment procedure does not alter the initial demand much. Fig. 2c shows that the initial estimate of the cost function is already very close to the ground truth and, as iterations progress, the estimates remain very close to the ground truth.



(a)

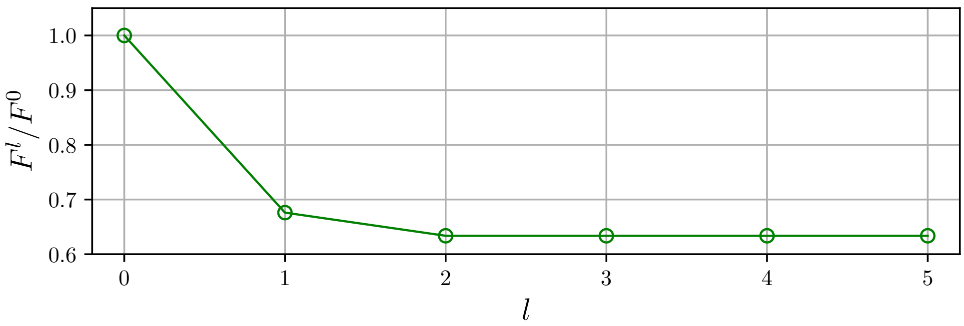


(b)

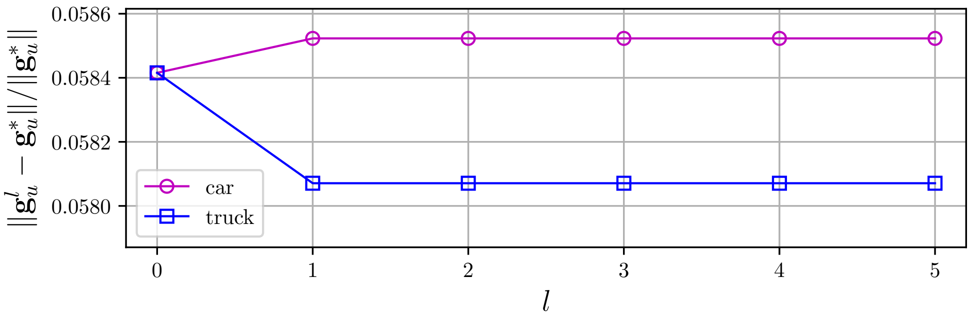


(c)

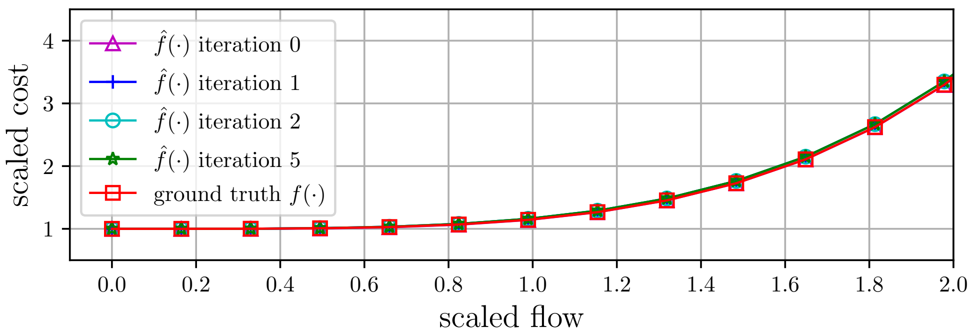
Figure 1: Key quantities and cost function estimations w.r.t. the joint iterations (Sioux-Falls; single-class): (a) normalized objective function values; (b) demand differences w.r.t. ground truth; (c) cost function estimates.



(a)



(b)



(c)

Figure 2: Key quantities and cost function estimations w.r.t. the joint iterations (Sioux-Falls; multi-class): (a) normalized objective function values; (b) demand differences w.r.t. ground truth; (c) cost function estimates.

4.2. Berlin-Tiergarten network

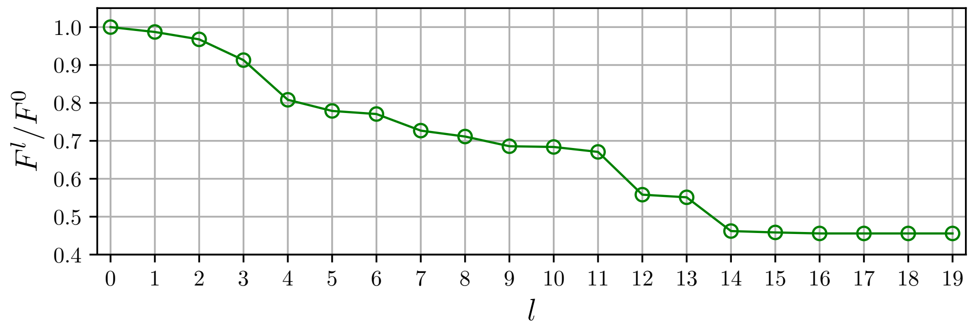
The Berlin-Tiergarten network [22] contains 26 zones (hence, $26 \times (26-1) = 650$ OD pairs), 361 nodes, and 766 links.

Single-class model

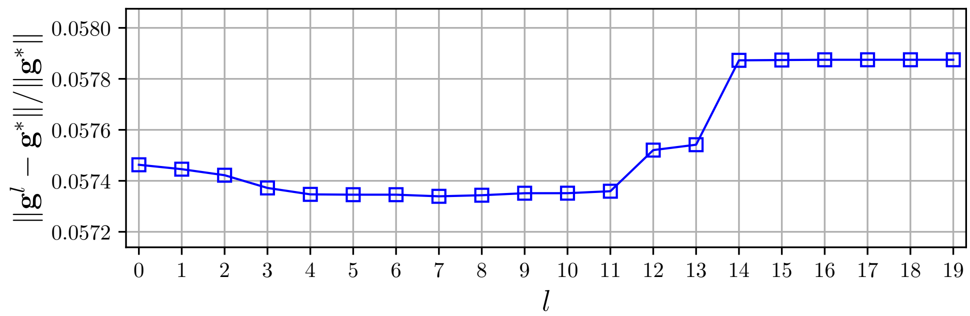
We use $n = 6$, $c = 0.5$, and $\gamma = 0.001$ in (15). See Fig. 3 for the results. Fig. 3a shows that, after 19 iterations, the objective function value of BiLev is reduced by more than 50%. Similar to the Sioux-Falls network single-class scenario, from Fig. 3b we see that the distance between the adjusted demand and the ground truth does not keep decreasing as iterations progress, but still, the distance changes very slightly, suggesting that the adjustment procedure does not alter the initial demand significantly. It is also seen from Fig. 3c that, as iteration count increases, the estimate of the cost function improves to some degree.

Multi-class model

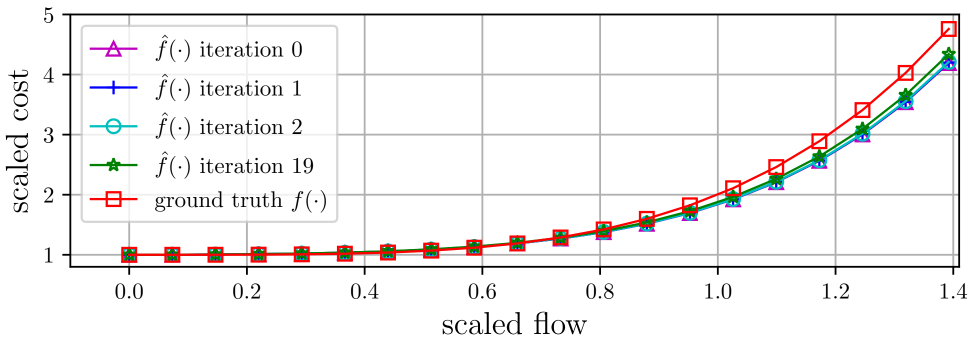
We use $n = 7$, $c = 1.5$, and $\gamma = 0.1$ in (15). See Fig. 4 for the results. Fig. 4a shows that, after 9 iterations, the objective function value of BiLev is reduced by more than 12%. Fig. 4b shows that, for either vehicle class (private car or commercial truck), the distance between the adjusted demand and the ground truth keeps decreasing and the distance changes very slightly, meaning the adjustment procedure does not alter the initial demand much. Similar to the Sioux-Falls network multi-class scenario, from Fig. 4c we see that, the initial estimate for the cost function is already very close to the ground truth, and as iterations advance, the estimates remain very close to the ground truth.



(a)

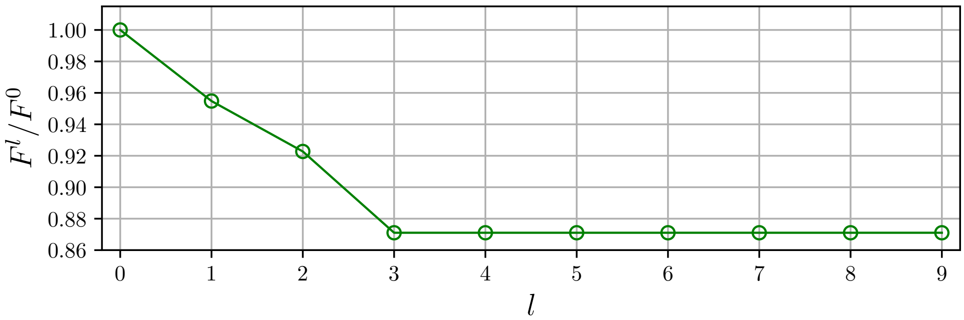


(b)

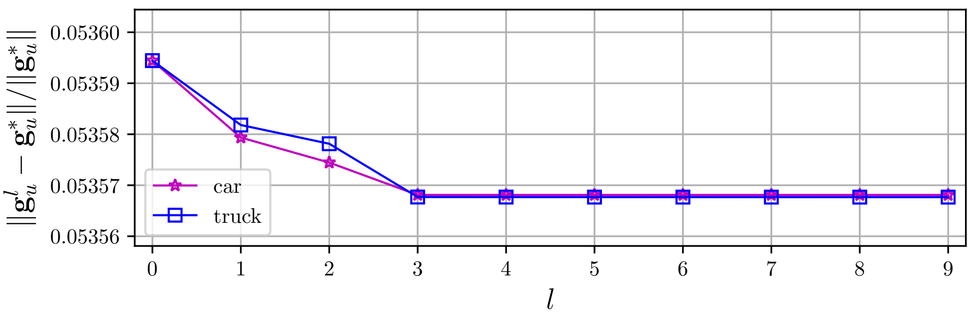


(c)

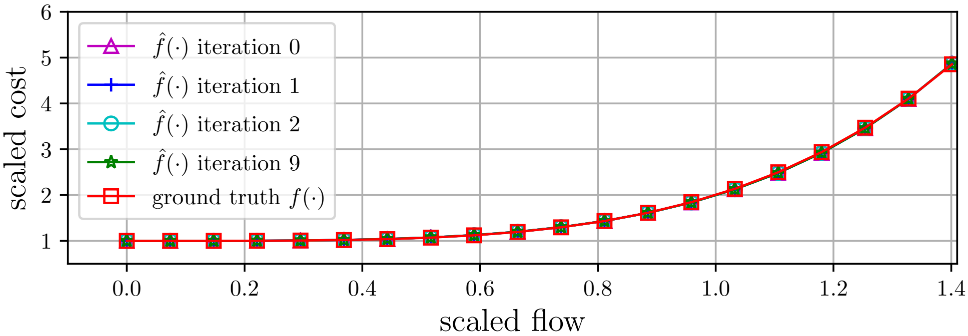
Figure 3: Key quantities and cost function estimations w.r.t. the joint iterations (Berlin-Tiergarten; single-class): (a) normalized objective function values; (b) demand differences w.r.t. ground truth; (c) cost function estimates.



(a)



(b)



(c)

Figure 4: Key quantities and cost function estimations w.r.t. the joint iterations (Berlin-Tiergarten; multi-class): (a) normalized objective function values; (b) demand differences w.r.t. ground truth; (c) cost function estimates.

4.3. Anaheim network

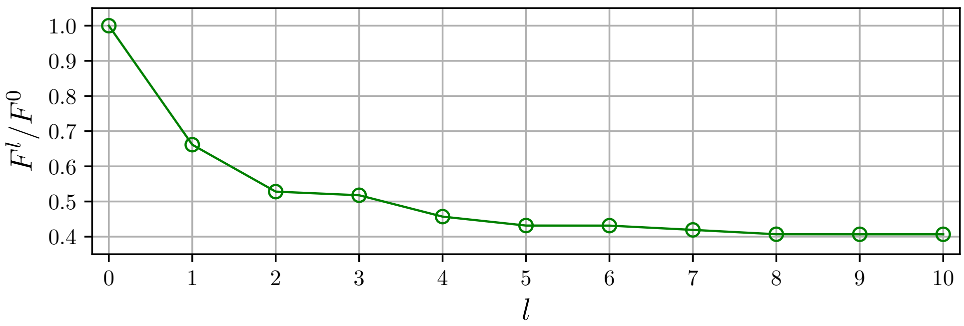
The Anaheim network [22] contains 38 zones (hence, $38 \times (38 - 1) = 1406$ OD pairs), 416 nodes, and 914 links.

Single-class model

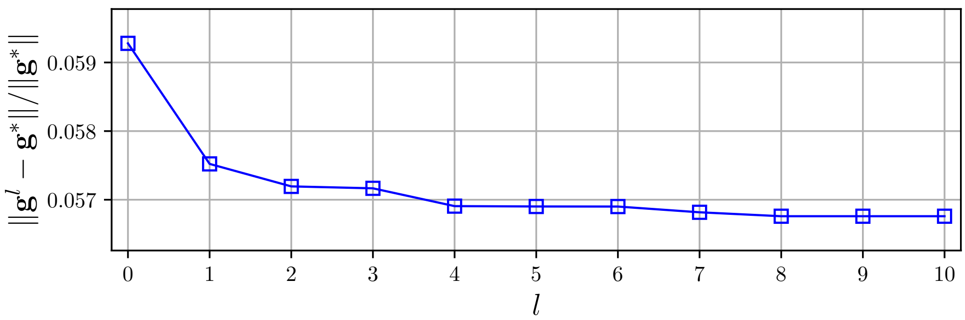
We use $n = 6$, $c = 3.5$, and $\gamma = 1.0$ in (15). See Fig. 5 for the results. Similar observations can be made as those for the Sioux-Falls network single-class scenario; a minor difference is that (see Fig. 5b) the distance between the adjusted demand and the ground truth demand keeps decreasing as the iteration count increases.

Multi-class model

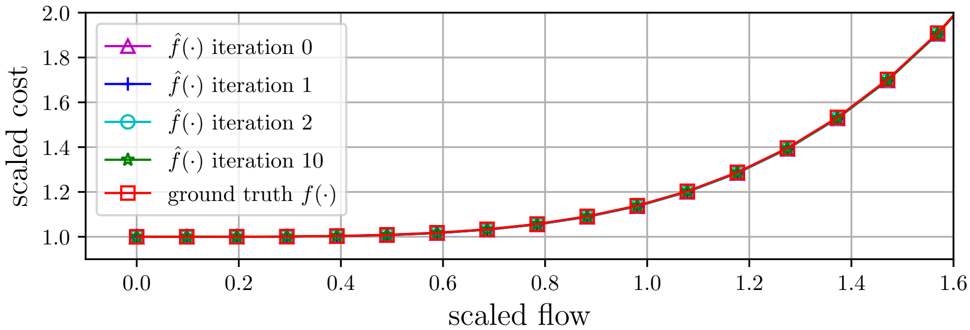
We use $n = 6$, $c = 1.5$, and $\gamma = 0.1$ in (15). See Fig. 6 for the results. Similar observations can be made as those for the Berlin-Tiergarten network multi-class scenario; a slight difference is that (see Fig. 6c), the initial estimate for the cost function is a bit far way from the ground truth, and this prevents the iterative cost function estimates produced by our gradient-based method from getting very close to the ground truth.



(a)

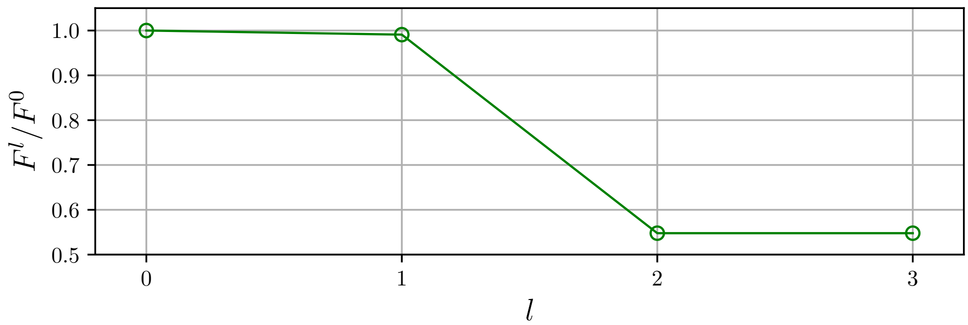


(b)

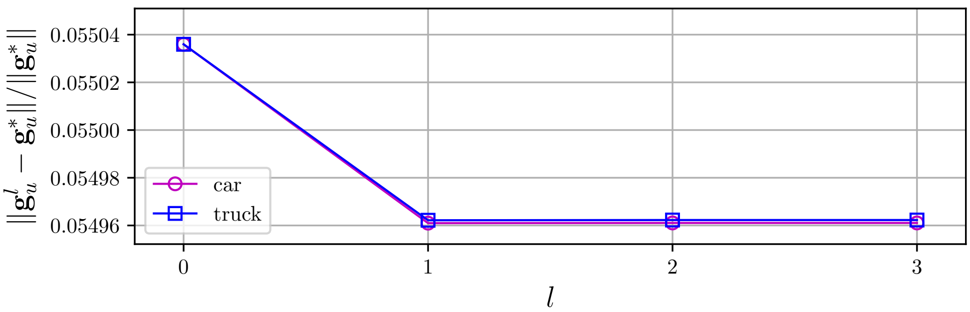


(c)

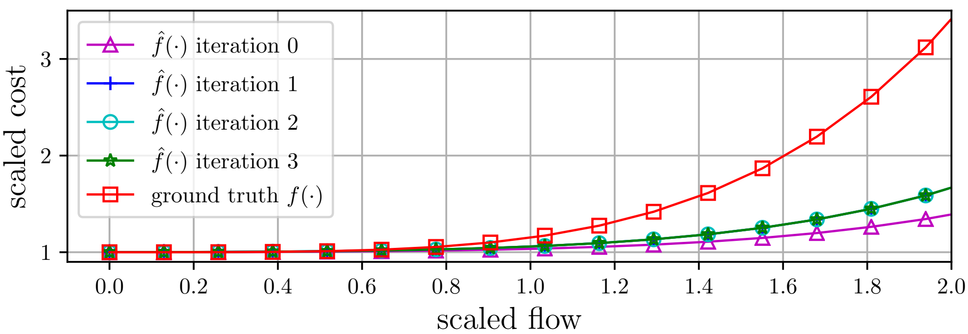
Figure 5: Key quantities and cost function estimations w.r.t. the joint iterations (Anaheim; single-class): (a) normalized objective function values; (b) demand differences w.r.t. ground truth; (c) cost function estimates.



(a)



(b)



(c)

Figure 6: Key quantities and cost function estimations w.r.t. the joint iterations (Anaheim; multi-class): (a) normalized objective function values; (b) demand differences w.r.t. ground truth; (c) cost function estimates.

Table 1: Summary of Sioux-Falls network single-class scenario experiments: (γ_1, γ_2) settings and key outputs (fix ground-truth $f(z) = 1 + 0.15z^4$ and set $(n, c, \gamma) = (6, 3.5, 1.0)$).

(γ_1, γ_2) **setting**	**iterations**	$F(\boldsymbol{\beta}, \mathbf{g})$ **reduction**	(γ_1, γ_2) **setting**	**iterations**	$F(\boldsymbol{\beta}, \mathbf{g})$ **reduction**
$(\mathbf{0}, 1)$	20	96.36%	$(\mathbf{1}, 1)$	20	81.94%
$(\mathbf{0.001}, 1)$	24	98.12%	$(\mathbf{10}, 1)$	11	45.72%
$(\mathbf{0.01}, 1)$	19	89.42%	$(\mathbf{100}, 1)$	3	13.15%
$(\mathbf{0.1}, 1)$	42	97.56%	$(\mathbf{1000}, 1)$	2	2.83%

Table 2: Summary of benchmark network experiments: sizes, parameter settings, and key outputs.

network	**zones/nodes/links**	**ground-truth** $f(\cdot)$	**model type**	(n, c, γ) **setting**	**iterations**	$F(\boldsymbol{\beta}, \mathbf{g})$ **reduction**
Sioux-Falls	24/24/76	$f(z) = 1 + 0.15z^4$	single-class	$(6, 3.5, 1.0)$	20	81.94%
			multi-class	$(6, 3.5, 1.0)$	5	36.63%
Tiergarten	26/361/766	$f(z) = 1 + z^4$	single-class	$(6, 0.5, 0.001)$	19	54.42%
			multi-class	$(7, 1.5, 0.1)$	9	12.89%
Anaheim	38/416/914	$f(z) = 1 + 0.15z^4$	single-class	$(6, 3.5, 1.0)$	10	59.33%
			multi-class	$(6, 1.5, 0.1)$	3	45.22%

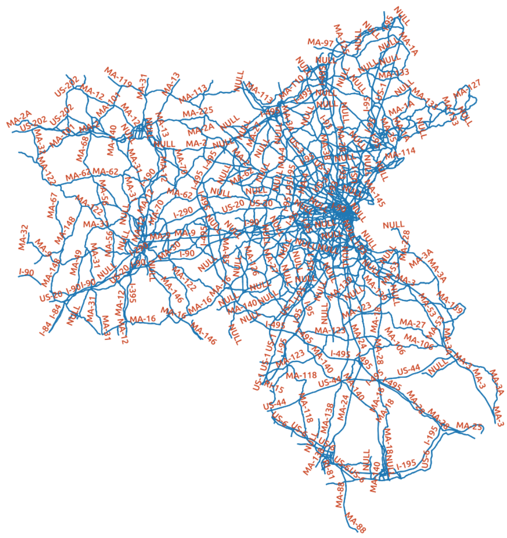


Figure 7: All available road segments in Eastern Massachusetts (from [8]).

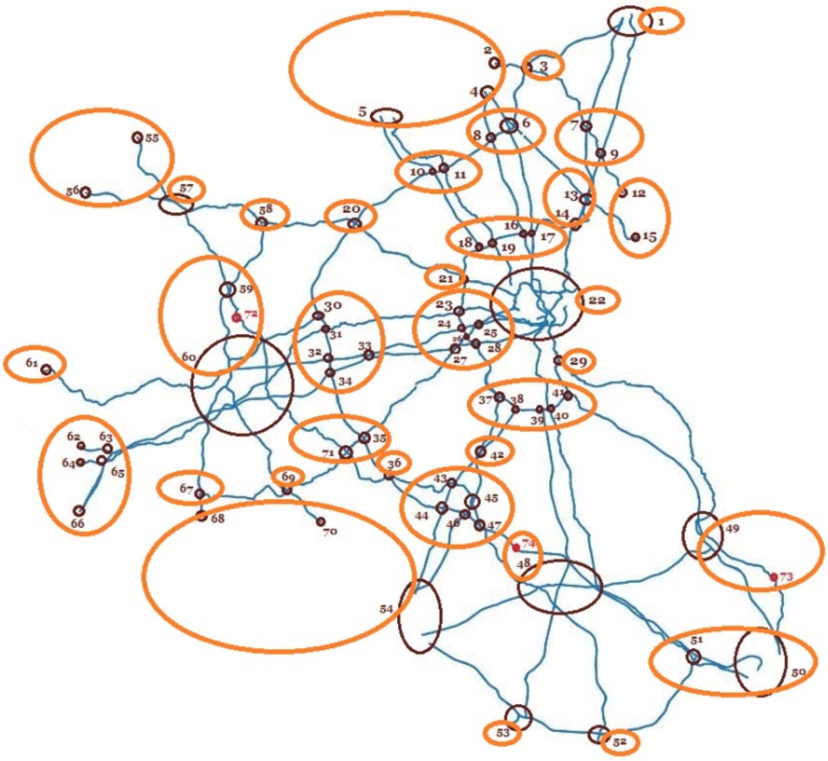


Figure 8: Eastern Massachusetts highway network; see [23] for the correspondences between nodes and link indices. ("nodes:zone" pairs – {1}: Seabrook (NH); {2, 4, 5}: NH; {3}: Haverhill; {6, 8}: Lawrence; {7, 9}: Georgetown; {10, 11}: Lowell; {12, 15}: Salem; {13, 14}: Peabody; {16, 17, 18, 19}: Burlington; {20}: Littleton; {21}: Lexington; {22}: Boston; {23, 24, 25, 26, 27, 28}: Waltham; {29}: Quincy; {30, 31, 32, 33, 34}: Marlborough/Framingham; {35, 71}: Milford; {36}: Franklin; {37, 38, 39, 40, 41}: Westwood/Quincy; {42}: Dedham; {43, 44, 45, 46, 47}: Foxborough; {48, 74}: Taunton; {49, 73}: Plymouth; {50, 51}: Cape Cod; {52}: Dartmouth; {53}: Fall River; {54, 68, 70}: RI; {55, 56}: VT; {57}: Westminster; {58}: Leominster; {59, 60, 72}: Worcester; {61}: Amherst; {62, 63, 64, 65, 66}: CT; {67}: Webster; {69}: Uxbridge.)

4.4. Eastern Massachusetts highway network

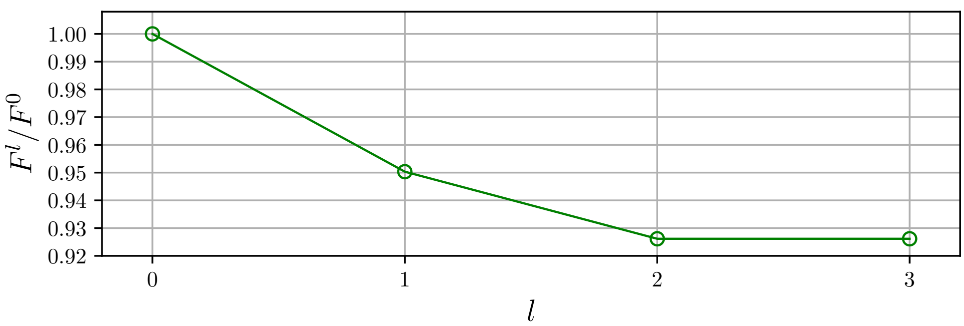
The Eastern Massachusetts (EMA) highway network contains 34 zones (hence, $34 \times (34-1) = 1122$ OD pairs), 74 nodes, and 258 links. We describe the EMA datasets and elaborate the data preprocessing procedures in the Appendix.

Single-class model

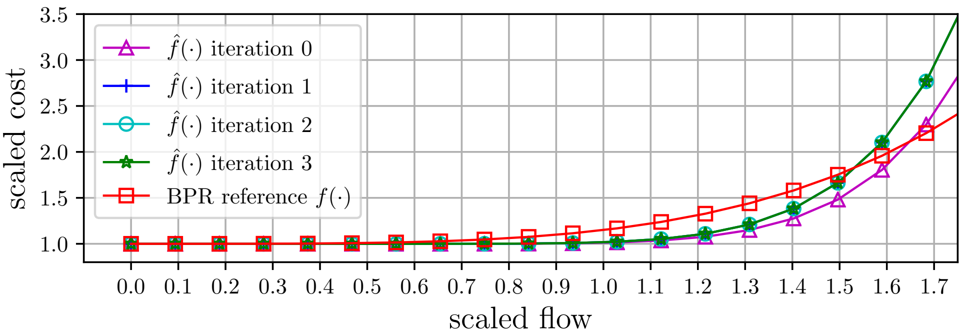
Take $n = 8$, $c = 1.5$, and $\gamma = 0.001$ in (15). See Fig. 9 for the results. Fig. 9a shows that, after 3 iterations, the objective function value of BiLev is reduced by more than 7%.

Multi-class model

Take $n = 8$, $c = 1.5$, and $\gamma = 0.001$ in (15). See Fig. 10 for the results. Fig. 10a shows that, after 7 iterations, the objective function value of BiLev is reduced by more than 5%.

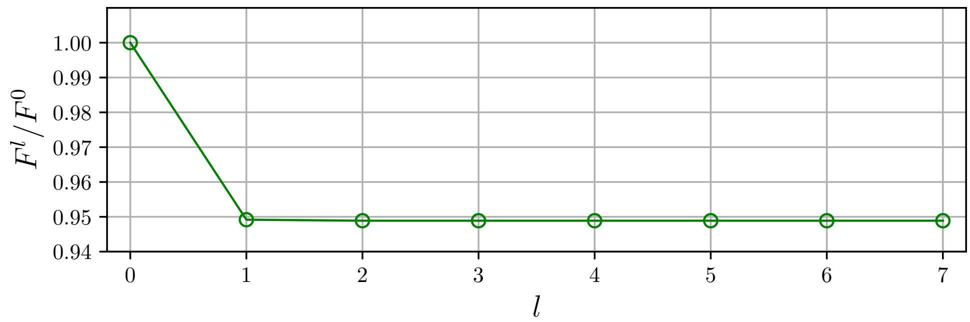


(a)

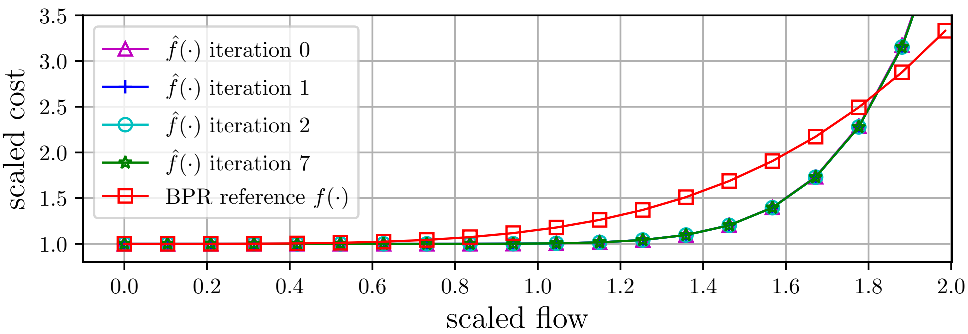


(b)

Figure 9: Key quantities and cost function estimations w.r.t. the joint iterations (EMA; single-class): (a) Normalized objective function values; (b) Cost function estimation.



(a)



(b)

Figure 10: Key quantities and cost function estimations w.r.t. the joint iterations (EMA; multi-class): (a) Normalized objective function values; (b) Cost function estimation.

Table 3: Summary of EMA highway subnetwork experiments: size, parameter settings, and key outputs.

network	**zones/nodes/links**	**reference** $f(\cdot)$	model type	(n, c, γ) **setting**	**iterations**	$F(\boldsymbol{\beta}, \mathbf{g})$ **reduction**
EMA	34/74/258	$f(z) = 1 + 0.15z^4$	single-class	$(8, 1.5, 0.001)$	3	7.38%
			multi-class	$(8, 1.5, 0.001)$	7	5.11%

5. Conclusions and future work

From the numerical results for benchmark networks we see that Alg. 1 (with Alg. 2 as a subroutine) works well in terms of reducing the objective function value of BiLev while improving the estimation accuracy for the cost function. Though we can not always expect that the two goals can be achieved simultaneously, our experience shows that we can always reduce the objective function value of BiLev to some extent, sometimes significantly; this is due to the construction of Alg. 1 (see Prop. 3.1). In addition, we have not observed cases where the cost function estimate deteriorates as iterations advance. Although we can not always expect the adjusted OD demands to be closer to the ground truth compared to their initial values, we see that in all our experiments, the actual adjustments on the initial demand are slight, meaning we do not alter the original data much. Furthermore, through the numerical experiments over the EMA highway network, we demonstrate how our proposed algorithms can be applied using real data.

Note that, of course, in practice, the output of our algorithm would heavily depend on the initial demand data as well as the accuracy of the flow observations. It is also worth mentioning that the two subroutines for solving the forward VI problem (Alg. 2) and the inverse VI problem (we solve (15) using the Gurobi solver) would largely determine the total running time. Our experience suggests that more accurate outputs of these two subroutines would typically increase the number of iterations in Alg. 1. Take the subroutine Alg. 2 for instance; in practice one may want to save some computation time by setting the parameter L (resp., ϵ_3) to a smaller (resp., bigger) value, thus sacrificing a little accuracy of the final OD demand vector and cost function.

Possible future directions include:

- As indicated in our earlier work [13], a potential more accurate estimate of the cost functions could be leveraged to facilitate smarter GPS navigation, thus reducing congestion (or, more specifically the "Price of Anarchy" [8, 9, 13]) in road networks. It is also possible to perform sensitivity analysis of link congestion metrics with respect to key quantities, such as link capacity and free-flow speed, similar to the analysis we have conducted in [13] for single-class networks. The results of this analysis would enable the Transportation department of a city to prioritize congestion-reducing interventions.

- One can consider integrating our algorithms to a dynamic OD demand estimation problem setting; see, e.g., [24], among others. The potential outcome would be to provide a more trustworthy database so as to predict the "Estimated Time of Arrival" [25] more accurately.

Appendix

We first provide a brief description of the datasets extracted from the Eastern Massachusetts (EMA) road network, and then elaborate on the data preprocessing.

I. Description of datasets

Thanks to the Boston Region Metropolitan Planning Organization (MPO), we have access to two datasets over the EMA road network: (i) The speed dataset includes the spatial average speeds (indistinguishable among different vehicle classes) for more than 13,000 road segments (with an average length of 0.7 miles; see Fig. 7) of EMA, providing the average speed for every minute of the year 2012. For each road segment, identified with a unique *tmc (traffic message channel)* code, the dataset provides information such as speed data (instantaneous, average, and free-flow speed) in *miles per hour (mph)*, date and time, and traveling time (in *minute*) through that segment. (ii) The flow capacity (in *vehicles per hour*) dataset includes capacity data for more than 100,000 road segments (with an average length of 0.13 miles) in EMA. For more detailed information of the two datasets, see [8].

II. Preprocessing

i) *Selecting a sub-network*

To reduce the computational burden while capturing the key elements of the Eastern Massachusetts road network, we only consider a representative highway sub-network as shown in Fig. 8, where there are 2984 road segments, composing a road network $\big(\tilde{\mathcal{V}}, \tilde{\mathcal{A}}, \tilde{\mathcal{W}}\big)$ with 74 nodes and 258 links. Further, we simplify the analysis by grouping nodes within the same area, assigning them the same *zone* label, thus obtaining 34 zones (as opposed to 74 nodes). Assuming that each zone could be an origin and a destination, then there are $34 \times (34 - 1) = 1122$ OD pairs. It is worth mentioning that the nodes 72, 73, and 74 are introduced for ensuring the identifiability of the OD demand matrices; see [26, Lemma 2]. We clarify here that, in our recent released EMA benchmark network [22], for simplicity we label each and every of the 74 nodes as a *zone*. However, only 34 of them play an actual role in composing OD pairs; to put it another way, there are $74 \times (74 - 1) - 34 \times (34 - 1) = 4280$ "fictitious" OD pairs whose flow demand is zero.

ii) *Calculating average speed and free-flow speed*

First, we select a time instances set $\mathcal{T}$ consisting of each minute of a PM peak period (5 pm – 7 pm) for each day of April 2012. Note that the selected PM period is a subinterval of the PM period in the capacity dataset. Then, we calculate the *average speed* for each road segment over $\mathcal{T}$ (covering 120 minutes). Finally, for each road segment, we compute a reliable proxy of the *free-flow speed* by using the 85th-percentile point of the observed speeds on that segment for all the time instances belonging to $\mathcal{T}$.

iii) *Aggregating flows of the segments on each link*

For $i \in [\![\tilde{\mathcal{A}}]\!]$, let $\{v_i^j, t_i^j, v_i^{0j}, t_i^{0j}, m_i^j; j = 1, \ldots, J_i\}$ denote the available observations (v_i^j, t_i^j), and parameters (v_i^{0j}, t_i^{0j}, m_i^j) of the segments composing the ith *physical* link, where, for each segment j, v_i^j (resp., v_i^{0j}) is the *average speed* (resp., *free-flow speed*; in *miles per hour*), t_i^j (resp., t_i^{0j}) is the *travel time* (resp., *free-flow travel time*; in *hours*), and m_i^j is the *flow capacity* (in *vehicles per hour*). Then, using Greenshield's model [27], we calculate the *traffic flow* (in *vehicles per hour*) on segment j by

$$\hat{x}_i^j = \frac{4m_i^j}{v_i^{0j}} v_i^j - \frac{4m_i^j}{(v_i^{0j})^2} (v_i^j)^2. \tag{25}$$

In our analysis, we enforce $v_i^j \leq v_i^{0j}$ to make sure that the flow given by (25) is nonnegative. In particular, if for some time instance $v_i^j > v_i^{0j}$ (this rarely happens), we set $v_i^j = v_i^{0j}$ in (25), thus leading to a zero flow estimation for this time instance. Aggregating over all segments composing link i we compute:

$$\hat{x}_i = \frac{\sum_{j=1}^{J_i} \hat{x}_i^j t_i^j}{\sum_{j=1}^{J_i} t_i^j},\ t_i^0 = \sum\nolimits_{j=1}^{J_i} t_i^{0j},\ m_i = \frac{\sum_{j=1}^{J_i} m_i^j t_i^{0j}}{\sum_{j=1}^{J_i} t_i^{0j}},$$

where $\hat{x}_i^j$ is given by (25) and $t_i^{0j} = v_i^j t_i^j / v_i^{0j}$, $j = 1, \ldots, J_i$.

iv) *Adjusting link flows to satisfy conservation*

For $i \in [\![\tilde{\mathcal{A}}]\!]$, let $\hat{x}_i$ denote the original estimate of the flow on link i (see the last step), x_i its adjustment, and ξ_{iu} the flow percentage on link i for vehicle class $u \in [\![\tilde{\mathcal{U}}]\!]$ (note that $\xi_{iu} \geq 0$ and $\sum_{u=1}^{|\tilde{\mathcal{U}}|} \xi_{iu} = 1$). Then, $x_{iu} = \xi_{iu} x_i$ (recall that x_{iu} denotes the flow on link $a(i, u)$; i.e., x_{iu} is the flow on link i for vehicle class u). We solve the following *Least Squares* problem:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \sum_{i=1}^{|\tilde{\mathcal{A}}|} \sum_{u=1}^{|\tilde{\mathcal{U}}|} \xi_{iu}^2 \left(x_i - \hat{x}_i\right)^2 && \qquad (26) \\ \text{s.t.} \quad & \sum_{i \in \mathcal{I}(v_j)} \xi_{iu} x_i = \sum_{i \in \mathcal{O}(v_j)} \xi_{iu} x_i, && \forall j \in [\![\tilde{\mathcal{V}}]\!],\ u \in [\![\tilde{\mathcal{U}}]\!], \\ & x_i \geq 0, && \forall i \in [\![\tilde{\mathcal{A}}]\!], \end{aligned}$$

where the first constraint enforces flow conservation for each node $v_j \in \tilde{\mathcal{V}}$ with $\mathcal{I}(v_j)$ (resp., $\mathcal{O}(v_j)$) denoting the set of links entering (resp., outgoing) to (resp., from) node v_j. Note that (26) generalizes its counterpart in [8, 9, 13]; the latter only tackles the case where $|\tilde{\mathcal{U}}| = 1$. For the case where $|\tilde{\mathcal{U}}| = 2$, the datasets available to us do not contain exact information of the parameters ξ_{iu}; in our experiments, we choose universally $\xi_{i1} = 0.8$ (for private cars) and $\xi_{i2} = 0.2$ (for commercial trucks).

v) *Estimating initial OD demand vectors*

For network $(\tilde{\mathcal{V}}, \tilde{\mathcal{A}}, \tilde{\mathcal{W}})$, to estimate initial OD demand vectors $\mathbf{g}_u = (g_{iu};\, i \in [\![\tilde{\mathcal{W}}]\!])$, $u \in [\![\tilde{\mathcal{U}}]\!]$, we borrow the Generalized Least Squares (GLS) method proposed in [26], which assumes the network to be uncongested (i.e., for each OD pair the route choice probabilities are independent of traffic flow), and the OD trips (traffic counts) to be Poisson distributed. In addition, we assume that the OD demands for different vehicle classes are independent from one another. Fix vehicle class $u \in [\![\mathcal{U}]\!]$. Denote by $\{\mathbf{x}_u^{(k)};\, k \in [\![\mathcal{K}]\!]\}$ $|\mathcal{K}|$ observations of the flow vector for vehicle class u. Let $\mathbf{S}_u = (1/(|\mathcal{K}| - 1)) \sum_{k=1}^{|\mathcal{K}|} \left(\mathbf{x}_u^{(k)} - \bar{\mathbf{x}}_u\right)\left(\mathbf{x}_u^{(k)} - \bar{\mathbf{x}}_u\right)'$ be the sample covariance matrix, where $\bar{\mathbf{x}}_u = (1/|\mathcal{K}|) \sum_{k=1}^{|\mathcal{K}|} \mathbf{x}_u^{(k)}$. Let $\mathbf{P}_u = \left[p_u^{ir}\right]$ denote the route choice probability matrix, where p_u^{ir} is the probability that a driver between OD pair i selects route r. Let $\boldsymbol{\xi}_u = \mathbf{P}'_u \mathbf{g}_u$. Then, the GLS method proceeds as follows:

(i) Identify a set $\mathcal{R}_u^i$ of feasible routes for each OD pair i, thus obtaining the link-route incidence matrix $\mathbf{A}_u$.

(ii) Solve sequentially the following two problems:

$$\text{(P1)} \quad \min_{\boldsymbol{\xi}_u \geq \mathbf{0}} \quad \frac{|\mathcal{K}|}{2} \boldsymbol{\xi}_u' \mathbf{Q}_u \boldsymbol{\xi}_u - \mathbf{b}_u' \boldsymbol{\xi}_u, \tag{27}$$

where $\mathbf{Q}_u = \mathbf{A}_u' \mathbf{S}_u^{-1} \mathbf{A}_u$ and $\mathbf{b}_u = \sum_{k=1}^{|\mathcal{K}|} \mathbf{A}_u' \mathbf{S}_u^{-1} \mathbf{x}_u^{(k)}$, and

$$\begin{aligned} \text{(P2)} \quad & \min_{\mathbf{P}_u \geq \mathbf{0},\, \mathbf{g}_u \geq \mathbf{0}} \quad h(\mathbf{P}_u, \mathbf{g}_u) \\ \text{s.t.} \quad & p_u^{ir} = 0 \quad \forall (i,r) \in \{(i,r) : r \notin \mathcal{R}_u^i\}, \\ & \mathbf{P}_u' \mathbf{g}_u = \boldsymbol{\xi}_u^0, \\ & \mathbf{P}_u \mathbf{1} = \mathbf{1}, \end{aligned} \tag{28}$$

where $h(\cdot, \cdot)$ can be taken as any smooth scalar-valued function, $\boldsymbol{\xi}_u^0$ is the optimal solution to (P1), and $\mathbf{1}$ denotes the vector with all 1's as its entries. Note that (P1) (resp., (P2)) is a typical *Quadratic Program (QP)* (resp., *Quadratically Constrained Program (QCP)*). Letting $(\mathbf{P}_u^0, \mathbf{g}_u^0)$ be an optimal solution to (P2), then $\mathbf{g}_u^0$ is our initial estimate of the demand vector for vehicle class u. It is worth mentioning that here we are able to deal with multi-class models, whereas in [8, 9, 13] only single-class models have been considered.

As pointed out in [13], we note that the GLS method would encounter numerical difficulties when the network size is as large as the one we are currently dealing with, because there would be too many decision variables in (P2). Thus, we perform a simplification procedure. In particular, we only consider the shortest route for each OD pair of $(\tilde{\mathcal{V}}, \tilde{\mathcal{A}}, \tilde{\mathcal{W}})$, thus, leading to a deterministic route choice matrix $\mathbf{P}$ and significantly reducing the number of decision variables in the QCP (P2).

For each vehicle class, we choose to estimate the initial OD demand vector corresponding to the selected PM peak period (5 pm – 7 pm) of April 2012, by leveraging $|\mathcal{K}| = 120$ samples of the flow vector. We take the average of these 120 flow vectors so as to obtain the observed flow vector $\mathbf{x}^*$.

Acknowledgments

The authors would like to thank the Boston Region MPO, and Scott Peterson in particular, for supplying the EMA data and providing us invaluable clarifications throughout our work.

References

[1] S. C. Dafermos, F. T. Sparrow, The traffic assignment problem for a general network, Journal of Research of the National Bureau of Standards B 73 (2) (1969) 91–118.

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

[3] J. G. Wardrop, Some theoretical aspects of road traffic research, Proceedings of the Institution of Civil Engineers 1 (1952) 325–378.

[4] P. Patriksson, The traffic assignment problem: models and methods, Utrecht, 1994.

[5] S. Dafermos, Traffic equilibrium and variational inequalities, Transportation science 14 (1) (1980) 42–54.

[6] D. Branston, Link capacity functions: A review, Transportation Research 10 (4) (1976) 223–236.

[7] D. Bertsimas, V. Gupta, I. C. Paschalidis, Data-driven estimation in equilibrium using inverse optimization, Mathematical Programming (2015) 1–39.

[8] J. Zhang, S. Pourazarm, C. G. Cassandras, I. C. Paschalidis, The price of anarchy in transportation networks by estimating user cost functions from actual traffic data, in: 2016 IEEE 55th Conference on Decision and Control (CDC), 2016, pp. 789–794. doi:10.1109/CDC.2016.7798364.

[9] J. Zhang, S. Pourazarm, C. G. Cassandras, I. C. Paschalidis, Data-driven estimation of origin-destination demand and user cost functions for the optimization of transportation networks, in: The 20th World Congress of the International Federation of Automatic Control, *accepted as Invited Session Paper*, arXiv:1610.09580, 2017.

[10] J. Zhang, I. C. Paschalidis, Data-driven estimation of travel latency cost functions via inverse optimization in multi-class transportation networks, in: 2017 IEEE 56th Conference on Decision and Control (CDC), *accepted*, arXiv:1703.04010.

[11] H. Spiess, A gradient approach for the OD matrix adjustment problem, Centre de Recherche sur les Transports, *Universite de Montreal (693).*

[12] J. T. Lundgren, A. Peterson, A heuristic for the bilevel origin–destination-matrix estimation problem, Transportation Research Part B: Methodological 42 (4) (2008) 339–354.

[13] J. Zhang, S. Pourazarm, C. G. Cassandras, I. C. Paschalidis, The price of anarchy in transportation networks: Data-driven evaluation and reduction strategies, Proceedings of the IEEE: special issue on "Smart Cities," submitted.

[14] Y. Noriega, M. A. Florian, Multi-Class Demand Matrix Adjustment, CIRRELT, 2007.

[15] M. Florian, Y. Chen, A coordinate descent method for the bi-level OD matrix adjustment problem, International Transactions in Operational Research 2 (2) (1995) 165–179.

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

[17] H. Yang, Q. Meng, M. G. Bell, Simultaneous estimation of the origin-destination matrices and travel-cost coefficient for congested networks in a stochastic user equilibrium, Transportation Science 35 (2) (2001) 107–123.

[18] Y. Noriega, M. A. Florian, Algorithmic approaches for asymmetric multi-class network equilibrium problems with different class delay relationships, https://www.cirrelt.ca/DocumentsTravail/CIRRELT-2007-30.pdf, Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT).

[19] S. Boyd, L. Vandenberghe, Convex optimization, Cambridge university press, 2004.

[20] T. Evgeniou, M. Pontil, T. Poggio, Regularization networks and support vector machines, Advances in computational mathematics 13 (1) (2000) 1–50.

[21] M. Patriksson, Sensitivity analysis of traffic equilibria, Transportation Science 38 (3) (2004) 258–281.

[22] Transportation Networks for Research, https://github.com/bstabler/TransportationNetworks.

[23] J. Zhang, S. Pourazarm, C. G. Cassandras, I. C. Paschalidis, InverseVIsTraffic, https://github.com/jingzbu/InverseVIsTraffic.

[24] A. R. Pitombeira-Neto, C. F. G. Loureiro, A dynamic linear model for the estimation of time-varying origin–destination matrices from link counts, Journal of Advanced Transportation.

[25] Estimated Time of Arrival, https://en.wikipedia.org/wiki/Estimated_time_of_arrival.

[26] M. L. Hazelton, Estimation of origin–destination matrices from link flows on uncongested networks, Transportation Research Part B: Methodological 34 (7) (2000) 549–566.

[27] T. Mathew, Traffic stream models, http://nptel.ac.in/courses/105101008/downloads/cete_03.pdf.