A congested schedule-based dynamic transit passenger flow estimator using stop count data

Qi Liu and Joseph Y. J. Chow

Department of Civil and Urban Engineering, New York University, Brooklyn, NY, USA

ABSTRACT

A network-level dynamic transit passenger flow estimation model based on congested schedule-based transit equilibrium assignment is proposed using observations from stop count data. A solution algorithm is proposed for the mathematical program with schedule-based transit equilibrium constraints with proven quadratic space- and time-complexities. The error bound is proven to be linearly proportional to the number of measurements. Computational experiments are conducted first to verify the methodology with two synthetic data sets (one of which is Sioux Falls compared to a benchmark model), followed by a validation of the method using bus data from Qingpu District in Shanghai, China from 1 July 2016, with 4 bus lines, 120 segments, and 55 bus stops. The estimated average of segment flows is only 2.5% off from the average of the observed segment flows; relative errors among segments are 42.5%, which fares well compared with even less complex OD estimation methods in the literature.

1. Introduction

Transit network information like time-varying origin-destination (OD) demand, passenger flows, and run ridership is indispensable for transit operations. Despite the importance of this information, many transit operators do not have access to data sources like smartcard or cellular phone that can be used to conveniently infer user trajectories hence transit run flows. This is possibly due to budget reason for small transit networks or privacy concerns.

Passenger flow estimation, i.e. of the ridership volumes on each run in the system, is a complex problem when Automated Passenger Counter (APC) use is limited. For one, the problem is generally more complex than other traffic flow estimation problems because of the added scheduling of transit lines, added walking, waiting, transfer activities, and limited data. It is clear there are two distinct classes of passenger estimation models: planning models and real-time operational models. The former use limited count data – turnstile or boarding/alighting counts and Automated Fare Collection (AFC) data – and seek to determine at least linked transit journeys (e.g. Gordon et al. (2013), Sun et al. (2015), Sun and Schonfeld (2015)). The results are used for strategic planning.

Transit operational prediction models, on the other hand, are applicable to many operational scenarios: station queue monitoring, incident management, holding and rescheduling strategies, and more (see Sun and Schonfeld (2015)). However, transit operational passenger flow estimation is more complex due to the transient nature of the problem. The presence of queueing in dynamic networks makes the passenger flow estimation a nonconvex problem. Unlike OD estimation, passenger flow
estimation is more complex. For example, such recent studies as Zhu, Koutsopoulos, and Wilson (2017) and Zhang et al. (2017) handle estimation at only either line or station level, not at network level. While other technologies exist to allow more sophisticated estimation using mobile communications data, e.g. Aguiléra et al. (2014) and De Regt et al. (2017), we focus on what can be estimated using only stop-based count data since many systems do not have access to the mobile device data.

We propose the first schedule-based transit passenger flow estimator at the network level using station count data, which includes estimating both OD demand and line flows. The estimation model can be used for evaluating operational strategies by outputting dynamic passenger flows. The main contributions of this paper are presented in bullets, first with this key contribution:

- This is the first study to propose a network-level passenger flow estimation model as a mathematical program with equilibrium constraints (MPEC) where the lower-level problem is a schedule-based transit assignment.

Related to this main contribution are three other contributions (that are only possible because of the first contribution):

- We propose two alternative solution algorithms, SDMSA and DSDMSA, to solve the MPEC and prove their space- and time-complexities are $O\left(\frac{N^2H^2L^2}{\delta \epsilon}\right)$ and $O(N^2H^2S\bar{L})$, respectively.
- The error bound for the model’s accuracy is proven to be linearly proportional to the number of measurements.
- We verify the model and algorithms using the toy Sioux Falls network against a benchmark algorithm and end up recommending SDMSA over DSDMSA, showing SDMSA is preferred. A comparison to a benchmark model without congestion consideration shows the proposed MPEC model reduces mean squared error (MSE) of the hourly OD by 83% and of the link flows by 92%. We further validate SDMSA on a real network: a local bus network in Shanghai involving 4 bus lines, 120 segments, and 55 bus stops, resulting in an average segment flow error of 2.6% and individual hourly segment flow error of 42.5%, which is in line in performance with transit OD estimation studies in the literature (e.g. Bierlaire and Crittin 2004; Cascetta, Inaudi, and Marquis 1993) despite the higher complexity of passenger flow estimation.

Section 2 elaborates on the research gap for a dynamic network-level passenger flow estimation model. Section 3 covers the methodology, which includes a new dynamic passenger flow estimation model as a mathematical program with schedule-based transit equilibrium constraints (MPEC), along with two solution algorithms. The equilibrium constraints corresponding to the schedule-based hyperpath flow are adapted from the literature with some modifications to fit it into an estimation problem. Properties of the MPEC and its solution algorithms are discussed. Section 4 covers computational experiments to verify the methodology with two synthetic data sets (one of which is Sioux Falls). Section 5 presents a validation of the method using data from Qingpu District in Shanghai, China. Section 6 concludes.

2. Literature review

Early literature on network estimation started with OD estimation without consideration of passenger assignment. Measurements are used as constraints; the goals include maximizing entropy (Macgill 1977), minimizing information (Willekens 1980), etc. In cases where measurements can be inconsistent, the goal is to maximize likelihood (Van Zuylen and Willumsen 1980) or minimize least squares in errors (Cascetta 1984). Time series tools have also been applied for OD flow estimation; Cremer and Keller (1987), Nihan and Davis (1987) have sought to use statistical techniques to estimate O/D matrices. Some online OD flow estimations based on Kalman filter have been proposed (Okutani...
and Stephanedes 1984; Ashok and Ben-Akiva 2000). Other methods include network tomography approaches (Zhang et al. 2017) and network-oriented machine learning like multi-agent inverse optimization (Xu et al. 2018). Kumar, Khani, and He (2018) used smart card data to infer passenger paths and OD matrices. An overview of transit OD estimation methods can be found in Hussain, Bhaskar, and Chung (2021).

User equilibrium (UE) assignment-based estimation models have been used to account for traveller behaviour due to limited data (Yang et al. 1992). In such models, OD flows are estimated such that their UE assignment best fits the count data. As suggested in recent studies (see Hussain, Bhaskar, and Chung 2021), however, transit OD estimation has sufficient data available now (e.g. APC, AFC, AVL) that data-driven methods can often be applied without relying on these traditional model-based methods. Still, for the passenger flow estimation problem, such data in practice can lead to stopping count data that nonetheless may be insufficient to predict unobservable flows. Because of these added complexities over OD estimation, such studies as Zhu, Koutsopoulos, and Wilson (2017) and Zhang et al. (2017) handle estimation at only either line or station level, not at network level.

To conduct passenger flow estimation at the network level using stop count data, we hypothesize that model-based approaches are still necessary. Such models combining estimation error minimization and UE condition have been considered for OD estimation (Yang et al. 1992), but calling it a proper bilevel problem, i.e. Stackelberg game (see Bard (2013)), does not make sense because the upper-level problem is not technically a separate decision-maker making changes to the network. It is rather an estimation problem with equilibrium constraints. Indeed, further study by Yang (1995) showed the estimation results using algorithms assuming a Cournot-Nash equilibrium versus those assuming a Stackelberg game are indistinguishable in performance of accuracy. Hence, the passenger flow estimation problem with equilibrium constraints can be considered an MPEC but not a bilevel problem.

As for models of the UE constraints, they are further classified into static UE and dynamic UE. Static models assume that the demand, service rate, and congestion level are constant throughout the analysis period. For transit, static UE typically refers to frequency-based models. The frequency-based approach assumes stochastic transit arrival times and that users choose from different common lines to form a preference set. It is typically assumed that users choose the first arriving line in the set, following the optimal strategy principle set by Spiess and Florian (1989). This leads to probabilistic paths, called hyperpaths, across multiple routes when the time dimension is ignored. The existence of asymmetric interactions leads to a variational inequality (VI) problem.

Transit assignment models can also be schedule-based (Wilson and Nuzzolo 2013; Chin, Lai, and Chow 2016). Schedule-based models assume that users choose from transit runs at transfer nodes. These available runs form an ordered set, sometimes called an ordered preference set; users always choose the most preferable run that is feasible for boarding. Whereas the frequency-based approach is simpler, schedule-based models are inherently dynamic; the evolution of the transit system state over time is tracked. Hamdouch and Lawphongpanich (2008), Hamdouch, Marcotte, and Nguyen (2004), Hamdouch et al. (2011), Nuzzolo, Crisalli, and Rosati (2012), Hamdouch, Szeto, and Jiang (2014) studied the dynamic UE for schedule-based transit assignment. In these models, rigid capacities are imposed and priorities among flows are considered. Some studies use simulation methods instead to evaluate within-day flow dynamics. Cats and West (2020) model the within-day dynamic path choice by means of day-to-day learning implemented in an agent-based simulation model.

It is apparent that passenger flow estimation for transit passengers needs to be dynamic for operational use. Wong and Tong (1998) proposed a maximum entropy estimator; Nuzzolo and Crisalli (2001) proposed a least-squares estimator; they both assume schedule-based transit service with no congestion effect. Postorino, Musolino, and Velonà (2004) compared OD estimation model under assumptions of frequency-based and schedule-based assignment for uncongested networks. Lam, Wu, and Chan (2003), Lam and Wu (2004), and Wu and Lam (2006) proposed least-squares OD estimators with frequency-based equilibrium assignment constraints with congestion effects. Montero,
Table 1. Summary of transit OD flow estimation models using count data.

| Studies                                | Uncongested (U) or Congested (C) | Frequency-based (F) Schedule-based (S) | (Upper-level) goals                      |
|----------------------------------------|-----------------------------------|---------------------------------------|-----------------------------------------|
| Nguyen, Morello, and Pallottino (1988) | U                                 | F                                     | Max entropy & max likelihood             |
| Wong and Tong (1998)                   | U                                 | S                                     | Max entropy                             |
| Nuzzolo and Crisalli (2001)            | U                                 | S                                     | Least squares                           |
| Postorino, Musolino, and Velonà (2004) | U                                 | F & S                                 | Least squares                           |
| Lam, Wu, and Chan (2003)               | C                                 | F                                     | Least squares                           |
| Lam and Wu (2004)                      | C                                 | F                                     | Least squares                           |
| Wu and Lam (2006)                      | C                                 | F                                     | Least squares                           |
| Montero, Codina, and Barceló (2015)    | –                                 | –                                     | Least MSE (Kaman Filtering)             |

Codina, and Barceló (2015) proposed real-time transit OD flow estimation formulated as Kalman filtering; proportions of passenger traversing links are calculated by assigning the most likely OD demand matrix; these proportions are updated when new measurements are available. The above models are all dynamic, meaning that they estimate time-dependent OD flows. There are studies on transit flow estimation using various data sources, like smartcard data (Pelletier, Trépanier, and Morency 2011; Tao, Rohde, and Corcoran 2014; Yang, Zhao, and Yao 2020) and cellular phone data (Sohn and Kim 2008; Caceres et al. 2012). In this study, we focus on estimation using the most basic type of data – count data. These studies are summarized in Table 1; none of these consider network-level passenger flow estimation.

As we can see, the literature in transit passenger flow estimation has not considered dynamic OD and passenger flows with congested schedule-based transit equilibrium constraints.

3. Proposed methodology

Our proposed model is a least squares estimation model where the underlying schedule-based transit equilibrium constraint is an extension of Nuzzolo and Crisalli (2001) model to congested schedule-based transit networks. Consider a transit network of a set of stops $N$ traversed by a set of lines $L$ over a time horizon $T$ set of intervals. A time-expanded (TE) network (Ahuja, Magnanti, and Orlin 1993) is used for the transit system representation, where each TE node is $i_h := (i, h) \in N \times T$. A hyperpath is a subgraph of the TE-network composed of an ordered preference set of attractive lines for each current passenger location $i_h$ to their destination. The measurements include entry, exit and pass-by counts at stops (via stop-based IoT sources like Wi-Fi). Count data are aggregated and reported periodically. The counting period of different measurements is assumed to be the same, although this can be trivially extended. The network and model representation are illustrated in Figure 1. It shows one hyperpath to destination node $N_5$ and how two users departing from $N_1$ at time 0 and 1 respectively get to destination $N_5$. Stop flows are reported every three time units at each stop.

**Notation**

| Parameters | Description |
|------------|-------------|
| $\mathcal{N}$ | Set of all physical transit stops; $N := \text{card}(\mathcal{N})$, the cardinality of $\mathcal{N}$, indexed by $i$; |
| $\mathcal{H}$ | Set of points in time; horizon $H := \text{card}(\mathcal{H})$, indexed by $h$; |
| $\mathcal{L}$ | Set of transit lines; $L := \text{card}(\mathcal{L})$, indexed by $l$; |
| $\mathcal{L}^\prime$ | the upper bound on the number of lines that any stop can have; |
| $\mathcal{K}$ | Set of measurements; $K := \text{card}(\mathcal{K})$; |
| $\mathcal{T}$ | Set of discrete time slices; $T := \text{card}(\mathcal{T})$ is the analysis horizon; |
| $\mathcal{W}$ | Set of tuples $(q, r, h)$ where $q \in N$ means the origin, $r \in N$ means the destination and $h \in T$ is the departure time; $W := \text{card}(\mathcal{W})$; |
| $\mathcal{S}_w$ | Set of hyperpaths for OD tuple $w \in \mathcal{W}$; |
| $\mathcal{S}$ | Set of all hyperpaths, namely $S := \bigcup_{w \in \mathcal{W}} \mathcal{S}_w$; |
| $\mathcal{\bar{S}}$ | the maximum number of hyperpaths for each $w$; |
Figure 1. Illustration of time-expanded network and terminology used.

\[ \text{d}^w \] Demand for \( w \in \mathcal{W} \);

\[ \mathbf{d} \] Demand vector; \( \text{dim}(\mathbf{d}) = \mathcal{W} \);

\( \mathcal{A} \) Set of links on TE-network; \( A = \text{card}(\mathcal{A}) \);

\( \mathcal{A}^+(i_h) \) set of arcs emanating from node \( i_h \);

\( t_a \) Travel time of link \( a \in \mathcal{A} \);

\( c_a \) Travel cost of link \( a \in \mathcal{A} \);

\( C(\cdot) \) Total cost travel cost of the network, a function of hyperpath flow vector \( \mathbf{f} \);

\( p_{i_h}^w \) The proportion of flow from OD \( w \in \mathcal{W} \) measured at stop \( i \) at \( h \);

\( \mathbf{P} \) the proportion matrix, whose elements are \( p_{i_h}^w \);

\( X_{i,h} \) The quantity of entry/exit/pass-by flow at stop \( i \in \mathcal{N} \) at time \( h \), an intermittent variable;

\( X_{i,k} \) \( k \)th aggregated measurement of entry/exit/pass-by flow at stop \( i \in \mathcal{N} \);

\( \Omega \) Set of all feasible hyperpath flow vectors.

**Notation**

\( f^s \) Flow for hyperpath \( s \in \mathcal{S}_W \);

\( \mathbf{f} \) hyperpath flow vector; \( \text{dim}(\mathbf{f}) = \text{card}(\Omega) \);

\( p_{i,h}^w \) The proportion of flow from OD \( w \in \mathcal{W} \) entering/exiting/passing-by stop \( i \) at time \( h \);

\( \zeta^{s,l,\tau,\alpha}_{i,h} \) The fractional hyperpath \( s \in \mathcal{S}_W \) flow appearing at stop \( i \) at time \( h \) with the arriving route being \( l \in \mathcal{L} \) and arriving time being \( \tau \in [0,h] \);

\( y^{s,l,\tau,a}_{i,h} \) The fractional hyperpath \( s \in \mathcal{S}_W \) flow passing link \( a \) with the arriving route being \( l \in \mathcal{L} \) and arriving time being \( \tau \in \mathcal{T} \);
3.1. Proposed dynamic UE assignment-based passenger flow estimator

A MPEC structure is adopted. The objective function (1a) is to minimize the squared errors of the flows at each stop. The intermittent variable $x_{i,h}$ represents entry, exit, or pass-by flow at stop $i$ at time $h$; $X_{i,k}$ is the measurement of the $x_{i,h}$ during the $k$-th period. $x_{i,h}$ is related to OD demand $d$ through the detection probabilities $P$. Ordinary least squares (OLS) is used to estimate the path flows.

Equation (1b) is used to describe the relationship between demand variables and intermittent variables. $P_{w}^{i,h}$, an element of $P$, refers to the proportion of flow from OD $w$ measured at stop $i$ at $h$. $P$ is determined by the lower level schedule-based UE model, formulated as variational inequality constraints shown in Equation (1c)–(1d).

$$
\min_{d} \sum_{i \in N, k \in K} \left( \sum_{h=h_{k-1}}^{h_{k}} x_{i,h} - X_{i,k} \right)^2
$$

$$
\sum_{w \in W} P_{w}^{i,h} d_{w}^{i} = x_{i,h}, \forall i \in N, h \in H
$$

Equilibrium constraints:

$$
C(f)(\tilde{f} - f)^T \geq 0, \forall \tilde{f} \in \Omega
$$

$$
\sum_{s \in S_w} f_{s} = d_{w}, \forall w \in W
$$

$\mathbf{d} \geq 0$

3.2. Solution algorithm for the MPEC

To solve this MPEC model, we adopt the link usage proportion-based algorithm from Yang et al. (1992), Yang and Bell (1998), by separating the problem into two subproblems: the primary estimation subproblem in Equation (1) without equilibrium constraints Equation (1c)–(1d) and with fixed proportions, and the equilibrium problem with fixed OD demand. A proportion represents the ratio of passengers for a particular OD flow passing through a particular link. The algorithm, which we call the Alternate Updating Method (AUM), is shown below.

ALGORITHM AUM (Alternate Updating Method)

Step 1: Initialize proportions $P^{(0)}$;
Step 2: Solve the estimation subproblem to obtain $d^{(n)}$ using $P^{(n-1)}$;
Step 3: Solve the equilibrium problem to obtain $P^{(n)}$ using $d^{(n)}$ by algorithm SDMSA;
Step 4: If convergence criterion is met, then stop; if not, go to step 2.

To solve the VI problem in Step 3 of AUM, a diagonalization method (Sheffi 1985) is typically applied. When interaction effects are fixed, the VI problem can be turned into an optimization problem. Then the Method of Successive Averages (MSA) can be used to find a local optimum (Powell and Sheffi 1982), although more efficient methods also exist, e.g. Liu, He, and He (2009). We propose combining the streamlined diagonalization method and MSA algorithm (SDMSA) for solving Step 3.

ALGORITHM SDMSA (Streamlined Diagonalization and MSA)

Step 3.1: Find an initial hyperpath flow;
Step 3.2: At step $n$, load the hyperpath flow onto the network using algorithm HFL in Section 3.3; update the transit link costs;
Step 3.3: Fix the costs of the links (not just the interaction effects), then find the descent direction hyperpath flow using algorithm FDD in the following;
Step 3.4: Apply an MSA step to update the hyperpath flow;
Step 3.5: If convergence criterion is met, then stop, return the proportions $P$ to upper level; if not, go to step 3.2.

Experiments suggest that the major progress of the lower level assignment model is made in first few iterations. It is natural then to consider returning the lower level proportions $P_{i,h}^w$ without convergence to the UE solution in the hope of speeding up AUM. This idea is similar to the idea of streamlined diagonalization for solving UE with interactions effects. We call this variant estimator a double-streamlined estimator (DSE) since a second ‘streamline’ is used to jump out of the equilibrium subproblem iterations. We call this variant algorithm DSDMSA. This alternative will be used as a benchmark to compare against the proposed algorithm. A significant improvement of SDMSA over DSDMSA would suggest the importance of the equilibrium subproblem.

3.3. Schedule-based UE assignment model

The schedule-based UE assignment model in Hamdouch and Lawphongpanich (2008) is adopted to solve Step 3.2.

There are two types of priority rules for transit flows to consider: continuance priority and FCFS priority. Continuance priority means that users already on the vehicle have the priority to continue their service; FCFS priority means that users who wait longer have the priority to board. These are taken into consideration.

The node in the time-expanded network is indexed by $i_h, i \in N, h \in \mathcal{H}$. The TE-network is acyclic and nodes can be arranged in topological & chronological (T&C) order. The tuple $(i_h, j_{h,t})$ refers to a link in the TE-network; it is a transit run segment if $i$ and $j$ are different and a waiting link otherwise. Let $z^{s, l, \tau, i_h}$ denote the fractional hyperpath $s \in S_w$ flow appearing at stop $i$ at time $h$ with the arriving route being $l$ and the arriving time being $\tau$. Similarly, let $y^{w, s, l, \tau, a}$ denote the fractional hyperpath $s \in S_w$ flow passing link $a$ with the arriving route being $l$ and the arriving time being $\tau$.

The hyperpath flow loading process in Step 3.2 is summarized by algorithm HFL from Hamdouch, Marcotte, and Nguyen (2004). The node flow $z$ and link flow $y$ are alternately updated. The algorithm HFL is listed for completeness. To sum up, the way to find and load a new hyperpath can be described as follows:

- At each (lower level) iteration $k$, current hyperpath flows are loaded onto the network through the HFL algorithm; nodes are visited in topological order and users are propagated forward.
- An optimal hyperpath (under the previous loaded flow) is generated by algorithm FDD (dynamic programming); nodes are visited in reverse topological order. Basically, the optimal hyperpath is the optimal way to send marginal flow to a destination. This mimics the way to find a shortest path through dynamic programming for a road traffic network.
- User demands are reassigned to include the new hyperpath (MSA step).

**Algorithm HFL (Hyperpath Flow Loading) (Hamdouch, Marcotte, and Nguyen 2004)**

**Step 3.2.1:** Start with the first node in T&C order;
**Step 3.2.2:** ($z^{s, l, \tau, i_h}$ update) for node $i_h$, collect flows from all incoming links and departure OD flows; departure flow is described as from ‘waiting line’ and $\tau = h$;
**Step 3.2.3:** ($y^{s, l, \tau, a}$ update for continuance priority flows) load the flows that can and choose to continue to ride; delete saturated links from all preference sets, then set $\tau = 0$;
**Step 3.2.4:** ($y^{s, l, \tau, a}$ update by FCFS rule) for every unsaturated outgoing link, compute the flow representing the first-choice-demand for this link arriving at $\tau$; find the link with the smallest remaining-capacity/first-choice-demand ratio, $\delta$, and this link is saturated first (Marcotte, Nguyen, and Schoeb 2004); then each outgoing link has proportion $\delta$ of their first-choice demands fulfilled;
If $\delta \geq 1$, all flows arriving at $\tau$ are assigned to their first choices; if $\tau = h$ and this is the last node, then stop; if $\tau = h$ and this is not the last node, set current node to be the next node in T&C order, go to step 3.2.2; else $\tau < h$, set $\tau \leftarrow \tau + 1$, then repeat Step 3.2.4;
• If $\delta < 1$, update outgoing links’ capacity, delete the saturated link from all preference sets; then repeat Step 3.2.4 with $\tau$ unchanged.

To find the descent direction (Step 3.3 of Algorithm AUM), dynamic programming (DP) is usually applied. Nodes of TE-network arranged in reverse T&C order can be used to label the stages. The decisions to be made are preference sets, $E^{s,l,r,ih}$, which are ordered sets of links ending at stop $i_h$ for users arriving at stop $i$ from bus line $l$ and time $\tau$ on hyperpath $s$. Changes are made from the DP method used in Hamdouch and Lawphongpanich (2008). For example, the preference set is identified by tuple $s,l,\tau,i_h$ instead by $(s,\tau,i_h)$ as in Hamdouch and Lawphongpanich (2008). Correspondingly, the state of DP is changed to be $(l,\tau)$ instead of $\tau$. We emphasize that the incoming line info $l$ should not be neglected in the identification of the preference set. When considering node (stage) $i_l$, a user arriving at stop $i$ at time $t$ by a transit line does not necessarily have to enjoy the continuance priority. For example, a user arriving by a transit line may transfer to another line to reach the destination.

To deal with this, we expand on the original model to add incoming line info $l$ into the state to handle the possibility of passengers alighting and transferring to another line. Also, in this study, the arrival time information $\tau$ is required for determining the optimal value function, instead of just being optional as stated in Hamdouch and Lawphongpanich (2008). Along with the change of state is the change of the optimal value function.

The optimal value function $\phi(l,\tau,i_h)$ is the expected cost from $i_h$ to destination. Two classes of users should be distinguished in the calculation of $\phi(l,\tau,i_h)$. The first class of users includes those who arrive at $i_h$ by the waiting line, or by transit lines that have reached their ends. These users have no option of travelling a line with continuance priority. Let $l_a$ be the bus line of an outgoing link $a = (i_h,j_h+t_a)$. The expected cost $\pi(l,\tau,a)$ of travelling by this outgoing link to a destination is calculated by Equation (2). If $l_a$ is a waiting line, then the arrival time will not change and stay at $\tau$; otherwise it is changed to $h + t_a$. The user preference set for node $i_h$ is just composed of outgoing links sorted by $\pi(l,\tau,a)$ values. Then $\phi(l,\tau,i_h)$ is the sum of $\pi$’s weighted by the link access probabilities, as shown by Equation (3). These link-access-probabilities $p(a)$ are found by sending an infinitesimal flow, called virtual flow (Marcotte, Nguyen, and Schoeb 2004), from $i_h$, to mix with flows from the loading step. The method to calculate $p(a)$ can be found in Marcotte, Nguyen, and Schoeb (2004). As shown above, given the state $(l,\tau)$, the decision at $i_h$ is irrelevant of the upstream decisions; dynamic programming applies here.

$$\pi(l,\tau,a) = \begin{cases} c_a + \phi(l_a,h + t_a,j_{h+t_a}), & \text{for } l_a \text{ being bus line} \\ c_a + \phi(l_a,\tau,j_{h+t_a}), & \text{for } l_a \text{ being waiting line} \end{cases}$$

for each $a = (i_h,j_h+t_a) \in A$

(2)

$$\phi(l,\tau,i_h) = \sum_{a \in A^+(i_h)} p(a) \pi(l,\tau,a)$$

(3)

where $t_a$: the travel time of this link;
$c_a$: is the travel cost;
$p(a)$: the probability of boarding link $a$;
$\pi(l,\tau,a)$: expected cost of travelling by link $a$ to destination;
$\phi(l,\tau,i_h)$ is the expected cost from $i_h$ to destination;
$A$: set of arcs;
$A^+(i_h)$: set of arcs emanating from node $i_h$.

The second class of users have the option to continue. Hence they need to make a decision of whether to continue to ride or alight. For the former case, the preference set just consists of $l$, and the
expected cost $\phi_{\text{continue}}$ is calculated by Equation (4). For the latter case, the preference set is composed of outgoing links sorted by $\pi$ values as before and the expected cost $\phi_{\text{alight}}$ is also calculated by Equation (2), (3). Note what is implied by Equation (3) in this case is that once a user alights his or her line, he or she needs to queue to board any line even if that line is the line where he or she was from. Finally, the expected cost $\phi(l, \tau, i_h)$ is given by Equation (5). And the preference set is decided correspondingly. The steps for finding the descent direction, with modifications described above, are listed in algorithm FDD for completeness.

$$\phi_{\text{continue}}(l, \tau, i_h) = \pi^l(\tau) = c_a + \phi(l, h + t_a, j_h + t_a), i_a = l$$

(4)

$$\phi(l, \tau, i_h) = \min\{\phi_{\text{continue}}(l, \tau, i_h), \phi_{\text{alight}}(l, \tau, i_h)\}$$

(5)

**ALGORITHM FDD** (Finding Descent Direction)

*Step 3.3.1:* Start with the first destination, $r$;

*Step 3.3.2:* Start with the first node in reverse T&C order; if it’s origin node ($r_h$), then the distance $\pi$ is zero; otherwise, set $\pi$ to infinity;

*Step 3.3.3:* At current node $i_h$, compute the cost $\phi$ of boarding each outgoing link for each arriving time $\tau$;

*Step 3.3.4:* For all possible states $(l, \tau)$, choose the corresponding way of finding the preference sets and optimal value function $\pi$;

*Step 3.3.5:* If $i_h$ is the last node in reverse T&C order and $r$ is the last destination, then stop; if $i_h$ is the last node and $r$ is not the last destination, then $r \leftarrow r + 1$, go to step 3.3.2; otherwise, set current node to be the next node, go to step 3.3.3.

The initial strategy for each destination $r$ can be found based on the nodes’ distances to $r$ under ‘free flow’ condition, where ‘free flow’ means no congestion effect or capacity constraint. At node $i_h$, all downstream nodes are sorted according to their distances to obtain the preference set. It’s also recommended to include all these links with finite distances into the preference sets to help ensure feasibility.

In summary, the schedule-based passenger flow estimation method is illustrated in a data flow diagram in Figure 2.

### 3.4. Solution existence, uniqueness and algorithm convergence

The model’s goal is to minimize a quadratic function, bounded below by zero; the feasible set of the lower level model is non-empty, closed, and bounded; hence the optimal solution of the MPEC always exists. Generally, an equilibrium in this problem is not guaranteed to be unique so the model is nonconvex. As such, our algorithm obtains a local optimum.

Relative gap is usually used for the convergence test of the lower level VI problem (Marcotte, Nguyen, and Schoeb 2004), as shown in Equation (6). $C(f)$ refers to the total cost of the current loaded strategy flows $f$; it can be obtained immediately after each loading step, where we have a vector $f = [f_1, f_2, \ldots, f_n, 0, \ldots, 0]$ after $n$ MSA steps. For the estimation subproblem, the average relative change (ARE) (Sheffi 1985) or maximum relative change (Yang 1995) can be used for variables of interest like OD flows or link flows. When the OD flow variables are chosen, ARE is calculated using Equation (7). The problem with ARE is that it can only be calculated when the denominator is non-zero. MSE can be used instead, as shown in Equation (8).

$$\frac{|C(f) - C(g)|}{C(f)} \leq \delta$$

(6)

$$\frac{1}{W} \sum_{w \in W} \frac{|d^{w(n+1)} - d^{w(n)}|}{d^{w(n)}} \leq \epsilon'$$

(7)
Figure 2. Data flow diagram of the proposed estimation method considering new (white) and existing (grey) algorithms from literature.

\[
\frac{1}{W} \sum_{w \in W} (d^{w(n+1)} - d^{w(n)})^2 \leq \epsilon''
\]

where \( C(f) \): the total cost of the current loaded strategy flows \( f \); \( W \): set of OD pairs with cardinality \( W \) and indexed by \( w \); \( d^{w(n)} \): OD demand of \( w \).

3.5. Computational and statistical considerations

The first consideration is the unit time interval. It should be small enough that all transit segment travel times and waiting times be expressed as integral multiples of this interval such that the TE-network is applicable. Passengers and vehicles arriving within the same interval are considered happening simultaneously. Passengers can transfer between simultaneously arriving vehicles; simultaneously arriving users have the same FIFO priority. To satisfy these requirements, the unit time interval is expected to be 1 min or even less.

The second consideration is about horizon \( H \) setting. It is good practice to reserve 30–60 min ahead of the targeted period for initial loading. Another 30–60 min trailing time is also desired, since passengers during this time may not be able to get to their destinations. As a result, the horizon would be 2 h or longer; it means 120 units if 1 min is selected as the unit interval.
Proposition 3.1: The space complexity and time complexity for the schedule-based assignment algorithm are $O(N^2 H^2 SL)$ and $O(N^2 H^2 L^2 L)$ respectively.

Proof: First, we discuss the space complexity. At first glance, the critical part is storing the node flow variable $z^i$, $r^j$, $l^k$, $h$ and link flow variable $y^o$, $k$, $l^i$, $r^j$, $s^h$, $\tau$, $a$. Since these two variables are in fact 7-dimensional ($q$, $r$, $h$, $s^h$, $l$, $\tau$, $i^h$). However, there is no need to store the whole $z$ and $y$ vector. We just need to store the ‘active’ ones – those who have been loaded user flows but have not been ‘excavated’. In the HFL process, only link flows need to be recorded for FDD for use later, which has the size of the number of links. The TE-network is typically sparse just like the physical network, so the number of links, $A$, is comparable to the number of nodes in expanded network, namely $A \approx NH$, where $N$ is the number of stops and $H$ is number of time intervals in horizon. This data are not space demanding. Instead, storing preference set is the most critical. It’s 5-dimensional – ($r$, $s^h$, $l$, $\tau$, $i^h$) and of size $N^2 L^2 SL$, where $S$ is the upper bound on number of strategies for each destination and $L$ is the upper bound on the number of lines at a node in TE network. $\tilde{S}$ and $\tilde{L}$ are reasonably bounded in practice. So, the number of stops and the number of time intervals (determined by horizon and the time granularity) decide the space complexity.

The computational time is proportional to the total number of stages multiply by the time spent at each stage. The number of stages is the number of destinations $N$ multiplied by number of nodes in time-expanded network $NH$, leading to $N^2 H$. At each stage of HFL algorithm, each incoming and outgoing arcs are checked for each strategy, hence the complexity is proportional to $\tilde{S}L$. For FDD stage, for each $(l, \tau)$, the optimal strategy need to be found. That means each outgoing arcs may be checked. The complexity is of order $HL^2$. Hence, the time complexity for assignment algorithm is $N^2 H(\tilde{S}L + HL^2)$. As we mentioned, $\tilde{S}$ and $\tilde{L}$ are of lower order compared to $H$; hence the complexity is $N^2 H^2 L^2$.

The number of lower- and upper-level iterations are expected to be proportional to the inverse of thresholds $\delta$ and $\epsilon$, respectively. Hence, the time complexity of the algorithm AUM is $O\left(\frac{N^2 H^2 L^2}{\delta \epsilon}\right)$. The space complexity of AUM is still $O(N^2 H^2 SL)$. Experimental results show that a computer with 128GB RAM and 2 quad-core CPU takes about 40 min to solve each lower-level assignment iteration in Python for a network with 60 stops, 5 transit routes and 120 min horizon, and tolerance being 0.005 for both subproblems. In most previous studies with schedule-based assignment, the models were applied to very small networks, say 5 nodes in Marcotte, Nguyen, and Schoeb (2004) and 6 nodes network with 6 time intervals in Hamdouch and Lawphongpanich (2008). Nuzzolo, Crisalli, and Rosati (2012) applied their schedule-based assignment model to a network with 11 traffic zones and 9 transit lines; their example has a horizon of 2 h divided into 5-min slices. Their assignment takes one hour on a personal computer (Intel Core 2 Quad CPU, 4 Gb RAM), but it was not an estimation model. Computation time for our algorithm can be improved significantly for implementation using more efficient languages like C++ and relaxing the tolerance to 1%, which Hamdouch and Lawphongpanich (2008) noted to be sufficient. Despite this, the computational time for large network (with ~ 500 stops) would still be formidable. We argue that schedule-based behaviours mostly appear in transit systems with infrequent services, which typically lies in small cities with modest size.

In the examples of this study, we set the unit time interval to be 1 min. Users may not be able to transfer between two buses arriving at the same stop in the same one-minute interval. This is indeed a problem for the dynamic assignment model using a time-expanded network. The events happening in the same time interval are regarded as ‘simultaneous’. This is a drawback of this general type of models. Finer time granularity would help to ease the problem, but the cost is the complexity. As we mentioned in the complexity analysis, cutting the unit time interval by half means that the space and computational time requirements would increase four times. We argue that for transit systems with infrequent service, operators would typically take transfer into consideration when the headway gap in the schedule is less than one minute. We have chosen one minute in this study, but it may not be the optimal choice in practice for every system. Practitioners are recommended to customize this setting.

The number of stops measured affects the error of the estimator. The source of error includes: (1) users may not strictly follow the assumptions of assignment; (2) measurements are liable to errors.
These factors are subject to randomness. The estimator variance of an over-determined problem usually has this relation:

\[
\text{estimator variance} = \frac{\text{population distribution variance}}{\text{sample size}}
\]

However, the flow estimation problem is under-determined; the estimation error is difficult to formulate since errors get passed in a complex way. We try to get an upper bound for the MSE. Let’s study a simpler linear problem instead:

\[(P)\]

\[
Ax = b \\
0 \leq x \leq u
\]

where \(x\) is a \(n\) dimensional vector to estimate; \(A\) is a \(n \times m\) coefficient matrix and \(n \geq m\) (under-determined); \(b\) is a random vector whose components have i.i.d. errors \(\varepsilon_i\).

**Proposition 3.2:** The upper bound for the MSE of problem (P) is bounded by a linear function of the number of measurements.

**Proof:** Let \(m\) be the number of measurements. The flows are bounded below by zero and above by some reasonable upper bound, say capacity \(u\). Hence the errors are also bounded, even if there are no measurements at all.

If there are no bounds on \(x\), we typically solve this by \(x = A^T (AA^T)^{-1} b\), where \(A^T (AA^T)^{-1}\) is the pseudo-inverse; this is the solution with the least \(L^2\)-norm. Here we still solve this problem by finding the least \(L^2\)-norm solution. After changing of variables \(x = P\tilde{x}\), we can transform the coefficient matrix \(A\) to column echelon form \(AP = [M:0]\); then the original problem becomes:

\[
[M:0]\tilde{x} = b \\
0 \leq \tilde{x} \leq P^{-1}u
\]

Suppose \(M\) is a non-degenerate \(m \times m\) matrix. Let \(\tilde{u}\) be the largest component of \(P^{-1}u\). The largest error of the least \(L^2\)-norm solution happens when the real value of the \(\tilde{x}_i\)'s corresponding to the zero column takes on the upper bound value (elements of \(P^{-1}u\)). The MSE of \(\tilde{x}\) satisfies Equation (9).

\[
MSE \leq \frac{1}{n} [(n - m)\tilde{u}^2 + 1^T \text{var}(M^{-1}b)]
\]

\[
= [\tilde{u}^2 + \frac{1^T \text{var}(M^{-1}b)}{n}] - \frac{\tilde{u}^2}{n}m
\]

\[
= [\tilde{u}^2 + \frac{1^T M^{-1} \text{var}(\varepsilon) M}{n}] - \frac{\tilde{u}^2}{n}m
\]

\[
= (u^2 + \sigma^2) - \frac{\tilde{u}^2}{n}m
\]

\[
= c_1 - c_2 m
\]

where we assume that the errors are i.i.d. with variance \(\sigma^2\). The total squared error of \(x\) is the same with \(\tilde{x}\) if we assume that the change of variables is an orthogonal transformation; the MSE of \(x\) also follows Equation (9). Hence the MSE of this simpler problem is bounded by a linear function of the number of measurements \(m\). We expect our much more complex non-linear under-determined flow estimation problem has a similar form of error bound.

The aim of this part is to give a taste of how errors would behave in a much simpler case. Practitioners can use this result to estimate the relationship between errors and number of measurements.
There are correlations between measures from adjacent stops. In real cases, the correlation coefficient of counts from two adjacent stops with the same set of stop-by transit lines could be as high as 0.9. This may shrink the size of the effective data, leading to larger errors. Overfitting should not be a big problem here since the problem is under-determined.

4. Computational experiments

Several computational experiments are conducted to verify the assignment algorithm, the estimation algorithm, and to validate the methodology using real data. The source code for models and data for examples are shared at BUILT@NYU (https://github.com/BUILTNYU/transit-flow-estimation/tree/master/schedule-based).

4.1. Dynamic UE assignment illustration

The example in this section is used to verify the schedule-based UE assignment algorithm. The transit network is shown in Figure 3. There are two bus lines $l_1$ and $l_2$. Vehicle capacity is 200 for both lines. There are 100 users for OD $(N1,N2,0)$ and 150 users for $(N5,N2,0)$. Suppose that waiting cost equals $(1+\delta)$ times of travel time, where $\delta$ is a small positive number; and assume that congestion effect exists. Hence users $N1 \rightarrow N2$ would rather wait at $N3$ for $l2$ than detour on a congested bus to $N4$ or $N5$. The 150 $N5 \rightarrow N2$ users would always take $l2$ line.

In the first iteration, 100 units of flow from $N1$ take $l1$ to $N3$, then wait until time 35 when the first $l2$ run pass by $N3$. But there are only 50 units of remaining capacity, the probability of boarding is 0.5. The rest wait for the next $l2$ run that arrives 10 min later. The optimal strategy is to divert $N1$ originated users to transfer at node $N4$ for continuance priority at $N3$. After one MSA step, 50 $N1$ originated users follow the initial strategy, and the other 50 users transfer at node $N4$.

To find the optimal strategy for the second iteration, we can see that at node $N4$ the virtual flow’s probability of boarding the first $l2$ run is still 1.0, since 50 users from $N1$ plus 150 users from $N2$ equals the capacity 200 and the virtual flow is infinitesimal. Hence the optimal strategy does not change.

In the third iteration, 2/3 of 100 $N1$ originated users are diverted to $N4$ and the probability of boarding the first $l2$ run becomes smaller than 1. The optimal strategy now diverts $N1$ users to transfer at further stop upstream, $N5$. The steps are shown in Figure 4. Users would detour upstream further and further to gain continuance priority to bring down their own costs, until there is no benefit to the action. This phenomenon is called a ‘detouring upstream phenomenon’ and is expected to happen in general schedule-based UE assignment models.

At the $n$-th iteration ($n > 3$), $1/n$ proportion of $N1$ originated users transfer at $N3$; $2/n$ proportion of $N1$ originated users transfer at $N4$; the rest of the $(n-3)/n$ proportion of the $N1$ users transfer at node $N1$.

![Figure 3. Transit network in example 1.](image-url)
Figure 4. Path flows realized from hyperpath flows over iterations; (a) 150 units N5→N2 flow choose path 5-4-3-2; 100 units N1→N2 flow choose path 1-3-2, half of which cannot board the first bus run at stop 3; (b) half of N1→N2 flow divert to path 1-3-4-3-2 to seize continuance priority over the rest half; (c) 2/3 of N1→N2 flow divert to path 1-3-4-3-2; (d) all N1→N2 flow divert to path 1-3-4-5-4-3-2 at equilibrium.

The UE solution in this simple example is letting almost all N1 users transfer at node N5. The total cost of all flows is $C_f = 50(55) + (50 + 150)45 = 11750$. At iteration $n > 3$, the total cost of the users’ optimal strategy is $C_g = 100(p)(45) + 100(1 - p)(55) + 150(45)$, where $p = 50/(100(n + 3)/3)$ is the probability of boarding the first run of l2 at N3. The relative gap is shown to be $|C_g - C_f|/C_f = 30/(235(n - 3))$. This gap decays at a speed of $O(n^{-1})$, i.e. the convergence rate is constant, although this is not guaranteed for MSA.

4.2. Benchmark model and performance metrics

For the verification experiment, the performances of the algorithms are compared in a controlled demand environment. A benchmark model is selected to quantify the benefit of the proposed model. Since the proposed model is the first to incorporate a schedule-based transit assignment to account for congestion effect in the estimation, a benchmark is chosen that ignores the congestion effect, i.e. essentially the upper-level model only as shown in Equation (10). The objective is to minimize the differences between the estimated and observed measurements, where the estimated values are simply determined from a set of k-shortest paths based on free-flow travel times. The k-shortest paths on the TE-network (which is loopless since each link moves up in time) are generated using Yen’s (1971) algorithm. The first set of constraints relates the path flows and OD flows. The second set of constraints represents transit segment capacity requirements to ensure that selected path flows remain feasible.

$$\min_{d,f} \| Ad - X \|_2$$

subject to:
\[
\sum_i f_i \delta_{i,w} = d_w, \forall w
\]
\[
\sum_i f_i' \delta_{i,s} \leq \text{caps}, \forall s
\]
\[
d_w, f_{h,t} \geq 0
\] (10)

where \(A\): a matrix that relates OD flows to stop measurements under a given set of paths; \(d_w\): Demand for OD pair \(w\); \(f_i\): flow on path \(i\); \(\delta_{i,w}\): path-OD incidence; \(\delta_{i,s}'\): path-transit segment incidence; \(\text{caps}\): capacity of transit segment \(s\).

The quantify the accuracy of the methods, we use MSE and average relative error (ARE) shown in Equations (11)–(13) as the performance indices.

\[
\bar{\mu} = \frac{1}{\#\text{Seg}} \sum_{(ij) \in N \times N} \sum_{h \in T} y_{s,l,r,(i_{h_{dn+h}},j)}^{j}
\] (11)

\[
\text{MSE} = \left( \frac{1}{\#\text{Seg}} \sum_{(ij) \in N \times N} \left( \sum_{h \in T} y_{s,l,r,(i_{h_{dn+h}},j)}^{j} - f_{(ij)} \right)^2 \right)^{1/2}
\] (12)

\[
\text{ARE} = \frac{1}{\#\text{Seg}} \left( \sum_{h \in T} y_{s,l,r,(i_{h_{dn+h}},j)}^{j} - f_{(ij)} \right) / f_{(ij)}
\] (13)

where \(f_{(ij)}\) is the inferred segment flow for segment \((i,j)\); \(\#\text{Seg}\) is the number of transit segments on the transit network.

4.3. OD demand and ridership estimation verification

This section tests the effectiveness of the estimation model to retrieve the OD flows and ridership given that the transit flows are indeed distributed according to a schedule-based UE assignment with some controlled noise.

The network used in this example is the Sioux Falls network shown in Figure 5 with three transit lines indicated in blue, green, and red. There are 14 stops on the bus lines. The time interval is 1 min and the horizon is 120 min. The congestion function is assumed to be of quadratic form: \(c = [(V/C)^2 + 1]t\), \(0 \leq V \leq C\), where \(V\) is the run flow; \(C\) is the capacity and \(t\) is the travel time. This function is convex; it takes on 1.0 at \(v/c = 0\) and 2.0 at \(v/c = 1.0\). The vehicle capacity is set to 100 and the bus headways are set to 15 min. The benchmark model uses \(k = 4\) shortest paths for each OD pair, resulting in 47,781 paths generated to assign the flows to minimize the error in Equation (10). The detailed settings can be found in the 4–2 example in the Github link provided earlier.

The measurement period is assumed to be the same as the time unit. The results represent the best possible estimation based only on the stop-level aggregated counting data. This establishes the upper bound for the performance of the proposed model. The demands are also loaded onto the network using the same schedule-based assignment model assumed by the estimation model. The ‘real’ link volumes and ‘observed’ measurement counts can be found in the 4–2 example in Github. The equilibrium subproblem’s threshold \(\epsilon\) is set to be 0.005. With this tolerance and software and hardware, each equilibrium subproblem iteration takes less than 5 min; a single estimation subproblem iteration takes about 30 min. We ran it up to 500 iterations without any stopping condition to observe the behaviour of the algorithm.
4.3.1. Model verification and comparison of SDMSA and DSDMSA

First, a comparison is conducted between the SDMSA (noted as the ‘standard estimator’ (SE)) and DSDMSA (noted as the ‘double-streamlined estimator’ (DSE)) algorithms to determine which is preferred. The demands are chosen to be spatially and temporally heterogeneous; the performances of estimators are shown in Figure 6. The trajectory of the objective value in Equation (1a) (SSE) is shown in Figure 7. The details of the estimation results of the two algorithms are shown from Table 2 to Table 4. The estimation for minute OD demand (N5→N15) is displayed in Table 2. The estimation results for hourly OD demands (N4, N5 originated) are shown in Table 3. Table 4 shows the transit run ridership estimation for southbound line l1. ‘Real’ refers to synthesized real data. MSE of SE is 0.076 for the minute-level OD demand, 156.7 for hourly OD demand, and 11.6 for transit link ridership.

The hourly OD estimation of SE is satisfactory. As we already know, the correlations between minute measurements can help us to better estimate OD demand compared to just using aggregated hourly count data. Suppose N stops all have measurements and there are H time points, then the data we have are \( N \times H \)-dimensional vectors, namely \( NH \) scalars in total. Suppose we want to estimate hourly OD demand which are linear combinations of minute demands, then the number of unknowns is \( N^2 \). As to minute level OD, as we can expect, it is not possible to estimate with accuracy, since the total
number of unknown is $N^2H$ – there is simply not enough information. As $N$ increases the number of minute OD variables grows at a rate $O(N^2H)$ while the number of measurements grows at rate $O(NH)$; this estimation problem would become more and more underspecified.

In both examples the convergence is not monotone. Furthermore, the performance of DSDMSA (DSE) is clearly not satisfactory. The MSE of the DSDMSA (DSE) is significantly larger than that of the standard one. SDMSA converges after 350 iterations; meanwhile, DSDMSA shows no sign of convergence. Therefore, the double-streamlined estimator is not recommended for use.

### 4.3.2. Comparison of the proposed model with the benchmark model

The comparisons of the proposed model under SDMSA with the benchmark model are shown in Table 5. MSE and ARE values are obtained at the minute OD flow, hourly OD flow, and combined flows across the links. The proposed model, obtained using the SE algorithm, performs significantly better, especially in terms of hourly OD flows (reduction to 17% of benchmark model’s MSE) and link flows.
Table 2. Minute OD demands estimation for pair N5-N21.

| h | Real | SE | h | Real | SE | h | Real | SE | h | Real | SE | h | Real | SE |
|---|------|----|---|------|----|---|------|----|---|------|----|---|------|----|
| 0 | 4    | 3  | 12| 4    | 2  | 24| 8    | 9  | 36| 8    | 7  | 48| 8    | 6  |
| 1 | 4    | 3  | 13| 4    | 2  | 25| 8    | 9  | 37| 8    | 7  | 49| 8    | 6  |
| 2 | 4    | 3  | 14| 4    | 2  | 26| 8    | 9  | 38| 8    | 7  | 50| 8    | 6  |
| 3 | 4    | 3  | 15| 4    | 2  | 27| 8    | 9  | 39| 8    | 7  | 51| 8    | 6  |
| 4 | 4    | 3  | 16| 4    | 2  | 28| 8    | 9  | 40| 8    | 7  | 52| 8    | 6  |
| 5 | 4    | 3  | 17| 4    | 2  | 29| 8    | 9  | 41| 8    | 7  | 53| 8    | 6  |
| 6 | 4    | 2  | 18| 4    | 2  | 30| 8    | 9  | 42| 8    | 7  | 54| 8    | 6  |
| 7 | 4    | 2  | 19| 4    | 2  | 31| 8    | 9  | 43| 8    | 7  | 55| 8    | 6  |
| 8 | 4    | 2  | 20| 8    | 4  | 32| 8    | 9  | 44| 8    | 7  | 56| 8    | 6  |
| 9 | 4    | 2  | 21| 8    | 9  | 33| 8    | 9  | 45| 8    | 7  | 57| 8    | 6  |
| 10| 4    | 2  | 22| 8    | 9  | 34| 8    | 9  | 46| 8    | 6  | 58| 8    | 6  |
| 11| 4    | 2  | 23| 8    | 9  | 35| 8    | 8  | 47| 8    | 6  | 59| 8    | 6  |

Table 3. Hourly OD demands estimation results for origins N4, N5.

| q | r | Real | SE | q | r | Real | SE | q | r | Real | SE |
|---|---|------|----|---|---|------|----|---|---|------|----|
| N5| N4| 0    | 2  | N5| N21| 400  | 334| N6| N11| 0   | 9  |
| N5| N5| 0    | 0  | N5| N22| 0    | 0  | N6| N14| 0   | 4  |
| N5| N6| 0    | 2  | N5| N23| 0    | 47 | N6| N15| 0   | 6  |
| N5| N8| 0    | 3  | N5| N24| 0    | 1  | N6| N16| 0   | 20 |
| N5| N9| 0    | 1  | N6| N4  | 0    | 0  | N6| N21| 0   | 54 |
| N5| N10| 0   | 1   | N6| N5  | 0    | 1  | N6| N22| 0   | 20 |
| N5| N11| 0   | 1   | N6| N6  | 0    | 0  | N6| N23| 400  | 264 |
| N5| N14| 0   | 2   | N6| N8  | 0    | 1  | N6| N24| 0   | 2  |
| N5| N15| 0   | 1   | N6| N9  | 0    | 0  | N6| N10| 0   | 9  |

Table 4. Transit run ridership estimation for south bound line l1.

| start | end   | Real | SE | start | end   | Real | SE | start | end   | Real | SE | start | end   | Real | SE |
|-------|-------|------|----|-------|-------|------|----|-------|-------|------|----|-------|-------|------|----|
| N4_0  | N5_5  | 0    | 0  | N10_45| N15_50| 100  | 100| N4_75 | N5_80 | 0   | 0  |
| N5_5  | N9_10 | 11   | 11 | N15_50| N22_55| 100  | 100| N5_80 | N9_85 | 0   | 3  |
| N9_10 | N10_15| 11   | 11 | N22_55| N21_60| 100  | 100| N9_85 | N10_90| 0   | 3  |
| N10_15| N15_20| 11   | 11 | N4_45 | N5_50 | 0    | 0  | N10_90| N15_95| 0   | 3  |
| N15_20| N22_25| 11   | 11 | N5_50 | N9_55 | 80   | 77 | N15_95| N22_100| 7   | 3  |
| N22_25| N21_30| 24   | 24 | N9_55 | N10_60| 80   | 76 | N22_100| N21_105| 12  | 30 |
| N4_15 | N5_20 | 0    | 0  | N10_60| N15_65| 100  | 85 | N4_90 | N5_95 | 0   | 0  |
| N5_20 | N9_25 | 46   | 46 | N15_65| N22_70| 100  | 100| N5_95 | N9_100| 0   | 0  |
| N9_25 | N10_30| 46   | 46 | N22_70| N21_75| 100  | 100| N9_100| N10_105| 0  | 0  |
| N10_30| N15_35| 46   | 46 | N4_60 | N5_65 | 0    | 0  | N10_105| N15_110| 0  | 0  |
| N15_35| N22_40| 46   | 46 | N5_65 | N9_70 | 61   | 60 | N15_110| N22_115| 0  | 0  |
| N22_40| N21_45| 64   | 64 | N9_70 | N10_75| 61   | 60 | N4_105| N5_110| 0   | 0  |
| N4_30 | N5_35 | 0    | 0  | N10_75| N15_80| 61   | 60 | N5_110| N9_115| 0   | 0  |
| N5_35 | N9_40 | 84   | 100| N15_80| N22_85| 73   | 70 | N9_100| N10_100| 100 | 100|

(reduction to 8% of benchmark MSE). It means that simple estimators like the benchmark model that do not use congestion effect to explain the passenger flows can underperform compared to a model with congestion effects.

4.3.3. Sensitivity analysis of measurement period and OD demand levels

The performances of the model under different measurement periods are tested and the results are shown in Figure 8. As the period becomes larger, the estimation errors for minute OD and ridership increase rapidly, although the hourly OD estimation does not seem to be influenced much. The error of ridership estimation soars when the period gets larger than 15 min. This suggests that the measurement period should be 15 min or less.
Table 5. Comparison of proposed model vs benchmark model.

| Performance index       | Ground truth | Benchmark model | Proposed model via SDMSA (% of benchmark) |
|-------------------------|--------------|-----------------|------------------------------------------|
| Total demand            | 800          | 1026            | 942 (62.8%)                              |
| Minute OD flow MSE      | –            | 0.173           | 0.076 (43.9%)                            |
| Minute OD flow ARE      | –            | 0.593           | 0.410 (69.1%)                            |
| Hourly OD flow MSE      | –            | 913.1           | 156.7 (17.2%)                            |
| Hourly OD flow ARE      | –            | 1.14            | 0.309 (27.1%)                            |
| Link combined flow MSE  | –            | 149.0           | 11.6 (7.8%)                              |
| Link combined flow ARE  | –            | 0.920           | 0.717 (77.9%)                            |

* ‘MSE’ means mean squared error; ‘ARE’ means avg. relative error.

Figure 8. Performance of standard estimator under varying measurement periods.

The estimation errors under different numbers of stops measured are shown in Figure 9. The robustness of this estimator with respect to link costs is tested by introducing link cost errors. The link cost (percentage) errors are generated by log-normal distribution. The estimation model is not aware of the errors. The estimation results are shown in Figure 10. The x-axis is the variance of log (percentage noise). When the variance is within 0.25, hourly OD demands and ridership estimation MSE lies within 20%. This result suggests that the estimation model is effective when the errors are within the interval $(e^{-\sqrt{1/4}}, e^{\sqrt{1/4}}) \approx (60\%, 160\%)$ with confidence 0.68 for this particular example.

To test the estimator performance with respect to demand patterns, we generate demand randomly with varying demand levels. The performances under these cases are summarized in Table 6. The error levels are similar to the previous results in Table 5. The SE model tends to overestimate the total demand. When the demand levels are lower, the errors performance are slightly better. For more detailed results, we refer to Github repository files.
Figure 9. Performance of standard estimator under varying number of stops measured.

Figure 10. Standard estimator performances under different link cost noise levels.
| #  | Network total Hourly OD demands | Estimated Total demands | Hourly OD demands | MSE   | ARE  |
|----|--------------------------------|-------------------------|------------------|-------|------|
| 1  | 429                            | 496                     | 67.06            | 0.41  |
| 2  | 863                            | 976                     | 142.56           | 0.32  |
| 3  | 1281                           | 1561                    | 253.20           | 0.58  |
| 4  | 1692                           | 1824                    | 402.13           | 0.61  |

5. Model validation with Shanghai bus data

5.1. Data for base scenario

We implement the model on a local bus network in Qingpu District, Shanghai, China to validate the estimation model and to show how it can be applied to transit operations. There are 4 bus lines, 120 segments, 55 bus stops, on this subnetwork; the bus routes are shown in Figure 11. The headways are between 5 and 10 min in the peak hour and between 10 and 20 min off-peak. When we consider one-minute time intervals across 60 min, this makes the problem significantly larger in size than any prior study of schedule-based transit assignment (which were limited to 6 nodes with 6 time intervals), and the first schedule-based passenger flow estimator based on a real data set.

We have 10GB of individual-level swipe data upon boarding collected from the network obtained between July 2016 to September 2016 from a Shanghai open data challenge (SODA, 2016). For that year, there were 15 million records for modes including bus, metro, and ferry. Out of that set, 5 million records are bus trips and 50,000 of those correspond to the Qingpu bus network evaluated in this case study. The recorded fields are card id, date, boarding time, line, mode, cost, and discount type. The stop info is not recorded for bus, but it can be partly extracted using the second-by-second boarding time data. User OD demands are then inferred by combining AM and PM boarding records, after which they are loaded onto transit segment flows by aggregating transit run flows based on their timestamps. For this experiment, we assume those inferred ODs are the ground truth corresponding to the AM peak hour (7:00–8:00AM) of Friday, July 1, 2016. Total AM peak hour demand is about 1,400. The inferred OD demands are illustrated by Figure 12.

Figure 11. Bus lines at Qingpu District, Shanghai, China.
Both the upper-level and lower-level thresholds $\varepsilon$ and $\varepsilon''$ are set to 0.005. Using those tolerances, each lower-level iteration takes about 38 min to run; a single upper level takes about 6 h on average; the estimation algorithm converged after 75 h. Since the code was written in the notoriously slow Python, it is possible to improve its run time for online implementation. For example, the algorithm can be made faster if JIT compiling is applied using PyPy or Numba or rewritten in Cython or C++. Increasing the tolerance to 0.01, as noted to be sufficient by Hamdouch and Lawphongpanich (2008), should cut the run time alone by a significant chunk. We can further reduce computational time by using 5-minute intervals instead of 1-minute intervals, which should then make the model operable online.

The convergence trajectory is shown in Figure 13. The algorithm is clearly convergent.

The test results are shown in Table 7 and visualized in Figure 14. The ‘inferred flow’ in (a) refers to the flows observed and inferred from the data; ‘estimated flow’ in (b) refers to the flow estimated using
Table 7. Estimator performance compared to inferred flow.

|                      | Observed mean $\mu$ | Estimated mean $\bar{\mu}$ | Std error (person) | ARE (%) |
|----------------------|---------------------|-----------------------------|--------------------|---------|
| Minute OD flow       | 0.05                | 0.06                        | 0.06               | 72.9%   |
| Hourly OD flow       | 3.26                | 3.39                        | 1.98               | 58.4%   |
| Segment flow         | 92.5                | 94.9                        | 33.1               | 42.50%  |

Figure 14. (a) Observed transit hourly flow inferred using individual level data; (b) hourly flow estimated by the model.

the proposed algorithm with the aggregated count data. The difference between the observed means and the estimated means are quite close, with the average of the segment flows only 2.6% off from the average of the observed flows. The standard error (0.06) and relative error (72.9%) are somewhat larger for minute-level OD flow but they decrease substantially when the flows are aggregated to hourly OD flows and distributed to segment flows; the relative error of the segment flow is 42.5%. By comparison, the OD estimation case study of Irvine network by Bierlaire and Crittin (2004) had relative mean errors in the range of 11% to 30% while Cascetta, Inaudi, and Marquis (1993) had a case study of dynamic OD estimation for the Brescia-Verona-Vicenza-Padua motorway in Italy with 19 centroids, 54 links, and 171 OD pairs, having relative mean errors of 17% to 49%. Considering our study deals with a more complex passenger flow estimation, our estimation is satisfactory at the segment flow level. The visualization of the observed and estimated flows shown in Figure 14 also suggests that the major segments are well-estimated.

5.3. Managerial/policy implications

With the model and algorithm validated, we now have a tool that can be applied to dynamic passenger flow estimation at the network level using stop count data that is becoming more readily available in public transit systems.
The dynamic aspect of the estimator makes it potentially applicable to transit incident management with some further improvements in computational efficiency. This would make it possible for network-level strategy deployment for incident management that has up to now eluded transit agencies because they are not able to effectively estimate passenger flows at the network level.

The model is also applicable to any schedule-based, fixed-route service system in which location data are available; this means other types of mobility services (route-based automated vehicle fleets, first/last mile hub networks, multimodal transit services) can benefit from this tool as well.

6. Conclusion
Transit flow estimation is important for transit planning and operations. In many transit systems, the only available information is the aggregated count data. This study focuses on the problem of dynamic transit flow estimation for schedule-based systems based on count data. Two key contributions are made.

First, we made modifications to the existing schedule-based assignment model to be more theoretically rigorous. Modifications on the definition of preference set and state, on the calculation of optimal value function, and on the decision of preference sets are proposed.

Second, we proposed two dynamic transit flow estimation methods (SDMSA and DSDMSA) based on the schedule-based user equilibrium concept. DSDMSA is a streamlined version of SDMSA and hence runs faster. The computational complexity is $O(N^2H)$ which makes it possible to apply the model in an online setting. Detouring upstream phenomena may occur on schedule-based transit networks.

The models are first tested on a small network. DSDMSA is shown to be ineffective while SDMSA is effective. The estimation errors (0.076 for the minute-level OD demand, 156.7 for hourly OD demand, and 11.6 for ridership) are satisfactory. The sensitivity of SDMSA to measurement period is also studied; the measurement period is recommended to be 15 min or less. Testing results for model robustness with respect to user perception show that the estimation model is effective when the perception errors are within the interval (60%, 160%).

The estimation model is tested using a realistic network. The estimated transit segment flow has a mean of 94.9 while the observed flow has a mean of 92.5 (difference of 2.5%); the standard error of estimation is 33.1 and the relative error is 42.5%.

For future research, this methodology may be extended to centralized time-dependent flow estimation for Mobility-as-a-Service systems. The stop count data can be considered with multiple other data sources. Also, this research may be used to study mobility providers’ decision to share information to get a better holistic picture. Lastly, real-time flow estimation methodology for non-schedule-based transit systems may be studied to complement this research.

Acknowledgements
Thanks are due to Haiyang Liu, Jian Wang and Chi Xie for sharing data with us. The authors are partially supported by the C2SMART University Transportation Center and U.S. National Science Foundation (NSF) CMMI-1652735.

Disclosure statement
No potential conflict of interest was reported by the author(s).

Funding
This work was supported by the U.S. National Science Foundation (NSF) [grant number CMMI-1652735]; U.S. Department of Transportation [grant number #69A3551747124].

ORCID
Qi Liu http://orcid.org/0000-0002-1587-8524
Joseph Y. J. Chow http://orcid.org/0000-0002-6471-3419
References

Aguilera, V., S. Allio, V. Benezech, F. Combes, and C. Milion. 2014. "Using Cell Phone Data to Measure Quality of Service and Passenger Flows of Paris Transit System." *Transportation Research Part C* 43: 198–211.

Ahuja, R. K., T. L. Magnanti, and J. B. Orlin. 1993. *Network Flows: Theory, Algorithms, and Applications*. New Jersey: Prentice-Hall.

Ashok, K., and M. E. Ben-Akiva. 2000. "Alternative Approaches for Real-Time Estimation and Prediction of Time-Dependent Origin–Destination Flows." *Transportation Science* 34: 21–36.

Bard, J. F. 2013. *Practical Bilevel Optimization: Algorithms and Applications*. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Bierlaire, M., and F. Crittin. 2004. "An Efficient Algorithm for Real-Time Estimation and Prediction of Dynamic OD Tables." *Operations Research* 52: 116–127.

Caceres, N., L. M. Romero, F. G. Benitez, and J. M. Del Castillo. 2012. "Traffic Flow Estimation Models Using Cellular Phone Data." *IEEE Transactions on Intelligent Transportation Systems* 13: 1430–1441.

Cascetta, E. 1984. "Estimation of Trip Matrices from Traffic Counts and Survey Data: A Generalized Least Squares Estimator." *Transportation Research Part B* 18: 289–299.

Cascetta, E., D. Inaudi, and G. Marquis. 1993. "Dynamic Estimators of Origin-Destination Matrices Using Traffic Counts." *Transportation Science* 27: 363–373.

Cats, O., and J. West. 2020. "Learning and Adaptation in Dynamic Transit Assignment Models for Congested Networks." *Transportation Research Record* 2674: 113–124.

Chin, A., A. Lai, and J. Y. J. Chow. 2016. "Non-additive Public Transit Fare Pricing Under Congestion with Policy Lessons from Toronto Case Study." *Transportation Research Record* 2544: 28–37.

Cremer, M., and H. Keller. 1987. "A new Class of Dynamic Methods for the Identification of Origin-Destination Flows." *Transportation Research Part B* 21: 117–132.

De Regt, K., O. Cats, N. Van Oort, and H. Van Lint. 2017. "Investigating Potential Transit Ridership by Fusing Smartcard and Global System for Mobile Communications Data." *Transportation Research Record* 2652: 50–58.

Gordon, J. B., H. N. Koutsopoulos, N. H. Wilson, and J. P. Attanucci. 2013. "Automated Inference of Linked Transit Journeys in London Using Fare-Transaction and Vehicle Location Data." *Transportation Research Record* 2343: 17–24.

Hamdouch, Y., H. Ho, A. Sumalee, and G. Wang. 2011. "Schedule-based Transit Assignment Model with Vehicle Capacity and Seat Availability." *Transportation Research Part B* 45: 1805–1830.

Hamdouch, Y., and S. Lawphongpanich. 2008. "Schedule-based Transit Assignment Model with Travel Strategies and Capacity Constraints." *Transportation Research Part B* 42: 663–684.

Hamdouch, Y., P. Marcotte, and S. Nguyen. 2004. "Capacitated Transit Assignment with Loading Priorities." *Mathematical Programming* 101: 205–230.

Hamdouch, Y., W. Y. Szeto, and Y. Jiang. 2014. "A New Schedule-Based Transit Assignment Model with Travel Strategies and Supply Uncertainties." *Transportation Research Part B* 67: 35–67.

Hussain, E., A. Bhaskar, and E. Chung. 2021. "Transit OD Matrix Estimation Using Smartcard Data: Recent Developments and Future Research Challenges." *Transportation Research Part C* 125: 103044.

Kumar, P., A. Khani, and Q. He. 2018. "A Robust Method for Estimating Transit Passenger Trajectories Using Automated Data." *Transportation Research Part C* 95: 731–747.

Lam, W., and Z. Wu. 2004. *Estimation of Transit Passenger Origin-Destination Matrices from Passenger Counts in Congested Transit Networks. Schedule-Based Dynamic Transit Modeling: Theory and Applications*. New York: Springer.

Lam, W. H., Z. Wu, and K. Chan. 2003. "Estimation of Transit Origin–Destination Matrices from Passenger Counts Using a Frequency-Based Approach." *Journal of Mathematical Modelling and Algorithms* 2: 329–348.

Liu, H. X., X. He, and B. He. 2009. "Method of Successive Weighted Averages (MSWA) and Self-Regulated Averaging Schemes for Solving Stochastic User Equilibrium Problem." *Networks* 9: 485.

Macgill, S. M. 1977. "Theoretical Properties of Biproportional Matrix Adjustments." *Environment and Planning A* 9: 687–701.

Marcotte, P., S. Nguyen, and A. Schoeb. 2004. "A Strategic Flow Model of Traffic Assignment in Static Capacitated Networks." *Operations Research* 52: 191–212.

Montero, L., E. Codina, and J. Barceló. 2015. *Dynamic OD Transit Matrix Estimation: Formulation and Model-Building Environment. Progress in Systems Engineering*. Dordrecht, The Netherlands: Springer.

Nguyen, S., E. Morello, and S. Pallottino. 1988. "Discrete Time Dynamic Estimation Model for Passenger Origin/Destination Matrices on Transit Networks." *Transportation Research Part B* 22: 251–260.

Nihan, N. L., and G. A. Davis. 1987. "Recursive Estimation of Origin-Destination Matrices from Input/Output Counts." *Transportation Research Part B* 21: 149–163.

Nuzzolo, A., and U. Crisalli. 2001. "Estimation of Transit Origin/Destination Matrices from Traffic Counts using a Schedule-based Approach." *Proc. AET Conference*, Cambridge, UK.

Nuzzolo, A., U. Crisalli, and L. Rosati. 2012. "A Schedule-Based Assignment Model with Explicit Capacity Constraints for Congested Transit Networks." *Transportation Research Part C* 20: 16–33.

Okutani, I., and Y. J. Stephanedes. 1984. "Dynamic Prediction of Traffic Volume Through Kalman Filtering Theory." *Transportation Research Part B* 18: 1–11.
Pelletier, M.-P., M. Trépanier, and C. Morency. 2011. “Smart Card Data use in Public Transit: A Literature Review.” *Transportation Research Part C* 19: 557–568.

Postorino, M., G. Musolino, and P. Velonà. 2004. *Evaluation of o/d Trip Matrices by Traffic Counts in Transit Systems. Schedule-Based Dynamic Transit Modeling: Theory and Applications.* Dordrecht, The Netherlands: Springer.

Powell, W. B., and Y. Sheffi. 1982. “The Convergence of Equilibrium Algorithms with Predetermined Step Sizes.” *Transportation Science* 16: 45–55.

Sheffi, Y. 1985. *Urban Transportation Networks.* NJ: Prentice-Hall, Englewood Cliffs.

SODA. 2016. Shanghai Open Data Apps. http://soda.data.sh.gov.cn/index_en.html, last accessed 4/2/22.

Sohn, K., and D. Kim. 2008. “Dynamic Origin–Destination Flow Estimation Using Cellular Communication System.” *IEEE Transactions on Vehicular Technology* 57: 2703–2713.

Spiess, H., and M. Florian. 1989. “Optimal Strategies: A New Assignment Model for Transit Networks.” *Transportation Research Part B* 23 (2): 83–102.

Sun, L., Y. Lu, J. G. Jin, D.-H. Lee, and K. W. Axhausen. 2015. “An Integrated Bayesian Approach for Passenger Flow Assignment in Metro Networks.” *Transportation Research Part C* 52: 116–131.

Sun, Y., and P. M. Schonfeld. 2015. “Schedule-based Rail Transit Path-Choice Estimation Using Automatic Fare Collection Data.” *Journal of Transportation Engineering* 142: 04015037.

Tao, S., D. Rohde, and J. Corcoran. 2014. “Examining the Spatial–Temporal Dynamics of Bus Passenger Travel Behaviour Using Smart Card Data and the Flow-Comap.” *Journal of Transport Geography* 41: 21–36.

Van Zuylen, H. J., and L. G. Willumsen. 1980. “The Most Likely Trip Matrix Estimated from Traffic Counts.” *Transportation Research Part B* 14: 281–293.

Willekens, F. J. 1980. “Entropy, Multiproportional Adjustment and the Analysis of Contingency Tables.” *Systemi Urbani* 2: 171–201.

Wilson, N. H., and A. Nuzzolo. 2013. *Schedule-based Dynamic Transit Modeling: Theory and Applications.* Dordrecht, The Netherlands: Springer Science & Business Media.

Wong, S., and C. Tong. 1998. “Estimation of Time-Dependent Origin–Destination Matrices for Transit Networks.” *Transportation Research Part B* 32: 35–48.

Wu, Z., and W. H. Lam. 2006. “Transit Passenger Origin-Destination Estimation in Congested Transit Networks with Elastic Line Frequencies.” *Annals of Operations Research* 144: 363–378.

Xu, S. J., M. Nourinejad, X. Lai, and J. Y. J. Chow. 2018. “Network Learning via Multiagent Inverse Transportation Problems.” *Transportation Science* 52: 1347–1364.

Yang, H. 1995. “Heuristic Algorithms for the Bilevel Origin-Destination Matrix Estimation Problem.” *Transportation Research Part B* 29: 231–242.

Yang, H., and M. G. H. Bell. 1998. “Models and Algorithms for Road Network Design: A Review and Some New Developments.” *Transport Reviews* 18: 257–278.

Yang, H., T. Sasaki, Y. Iida, and Y. Asakura. 1992. “Estimation of Origin-Destination Matrices from Link Traffic Counts on Congested Networks.” *Transportation Research Part B* 26: 417–434.

Yang, T., P. Zhao, and X. Yao. 2020. “A Method to Estimate URT Passenger Spatial-Temporal Trajectory with Smart Card Data and Train Schedules.” *Sustainability* 12: 2574.

Yen, J. Y. 1971. “Finding the k Shortest Loopless Paths in a Network.” *Management Science* 17 (11): 712–716.

Zhang, J., D. Shen, L. Tu, F. Zhang, C. Xu, Y. Wang, C. Tian, X. Li, B. Huang, and Z. Li. 2017. “A Real-Time Passenger Flow Estimation and Prediction Method for Urban Bus Transit Systems.” *IEEE Transactions on Intelligent Transportation Systems* 18: 3168–3178.

Zhu, Y., H. N. Koutsopoulos, and N. H. Wilson. 2017. “A Probabilistic Passenger-to-Train Assignment Model Based on Automated Data.” *Transportation Research Part B* 104: 522–542.