A TWO-STAGE STOCHASTIC PROGRAMMING APPROACH FOR DYNAMIC OD ESTIMATION

Qing-Long Lu
Chair of Transportation Systems Engineering, Technical University of Munich, Germany
Email: qinglong.lu@tum.de

Moeid Qurashi, Corresponding Author
Chair of Transport Modeling and Simulation, Technical University of Dresden, Germany
Email: moeid.qurashi@tu-dresden.de

Constantinos Antoniou, Ph.D.
Chair of Transportation Systems Engineering, Technical University of Munich, Germany
Email: c.antoniou@tum.de

Word Count: 5770 words + 0 table(s) $\times$ 250 = 5770 words

Submission Date: October 9, 2022

ABSTRACT

Estimating origin-destination (OD) demand is indispensable for urban transport management and traffic control systems. While the existing estimation methods rely on data sources like household travel surveys and traffic network detection, they incur very high costs and are still either less frequent or low in coverage density triggering lower observability and indeterminacy issues for OD estimation. With ubiquity of smartphones, Location based social networks (LSBN) data has emerged as a new rich data source with broad urban spatial and temporal coverage highly suitable for OD estimation. However, thus far, most LSBN-based estimation models only focus on static (day-level) OD estimation. This paper establishes a two-stage stochastic programming (TSSP) framework integrating the activity chains to model activity-level mobility flows using LBSN data. The first stage model aims to minimize the errors introduced by the inter-zone OD flows alongside the expected errors of the check-in patterns. The second stage model attempts to minimize the errors produced by the considered check-in pattern scenarios. A generalized Benders decomposition algorithm is presented to solve the two-stage stochastic programming model. We conduct the experiments employing generalized least squares (GLS) estimator on the case study of Tokyo city. The results depict that the algorithm convergence can be guaranteed within several steps. The algorithm shows satisfactory performance in check-in pattern estimation, OD flows estimation, and activity share estimation. Further, the implementation of the model in practical applications is also specifically discussed.

Keywords: OD estimation, Demand estimation, Large networks, Stochastic programming, LBSN data, Generalized Benders Decomposition

INTRODUCTION

An accurate origin-destination (OD) matrix, as a typical representative of mobility demand pattern, is indispensable in urban transportation management and traffic control systems (*1, 2*). Integrating with a traffic assignment model, it can be used to reproduce the traffic flow and network state in a detailed manner, helping to design practical measures to improve the transport system efficiency.

Existing OD estimation methods primarily includes three travel data sources, i.e., traditional household surveys, traffic measurements, and positioning technology based data (*3*). The first data source type of traditional household surveys are time-consuming, labor-intensive and expensive, therefore are normally restrained within a limited area at low frequencies (e.g., once or twice a decade). Similarly, OD estimation using the second data type of traffic measurements also relies on dense detection infrastructure distributed over the network, requiring high cost for both installation and maintenance. The traffic measurement based methods also structurally suffer from the issue of indeterminateness in estimating realistic OD flows patterns (i.e., multiple sets of varying OD matrix patterns can satisfy the constraints imposed by the traffic measurements and optimize the system objective at the same time) due to the high dimensionality of the OD matrix (*4–6*). Therefore, methods using the third data source type have attracted much attention in recent years. The ubiquity of smartphones equipped with positioning technologies, such as GPS and Bluetooth, has resulted in regular real-time generation of large sets of well-distributed data that also provides unprecedented opportunities for the implementation and application of OD estimation methods.

Generally, people travel for specific purposes, which enables the integration of travel purposes and activity chains into OD estimation. Location based social networks (LBSN) data has been used to develop such models, attributed to its broad urban spatial and temporal coverage and confirmed trip purposes (*7*). LBSN services generate a large amount of anonymous check-in data of venues and users, making it a natural "host" of urban mobility patterns (*3, 8*). More specifically, check-in time series of venues record the travel destination distribution in both spatial and temporal dimensions, while check-ins of users reflect the activity chains of individuals. It is thus probable to develop an activity-based OD estimator to simulate the urban mobility using the patterns extracted from LBSN data.

To this end, Yang et al. (*3*) proposed a singly constrained gravity model to estimate the non-commuting OD flows using LBSN check-in data. The model was further improved by Jin et al. (*8*) who replaced the singly constrained gravity model with a doubly constrained one to reduce the sampling bias of check-in data. The performance of other conventional trip distribution models, such as radiation model, rank-based model, and population-weighted opportunities model, calibrated with LBSN data have also been compared and evaluated in Kheiri et al. (*9*). However, all these models can only provide a static (day-level) solution to the OD estimation problem. Accordingly, inspired by the promising performance of the application of the Hawkes process to self-reinforcing behavior modeling in Cho et al. (*10*), Hu and Jin (*7*) presented a time-of-day zonal arrival estimation model by integrating the Hawkes process and a LBSN check-in observation model into a state-space modeling framework. Such an approach can reduce the sampling bias in OD estimation caused by the difference between the social behaviors and the real travel patterns. As per Jin et al. (*8*), the accuracy of trip arrival estimation is significant for the performance of OD estimator using LBSN data.

There is still a significant scope left to explore the usage of LBSN data and construct a dynamic OD estimator that can thoroughly utilize the trip purposes and activity chains information. Therefore, as a step in that direction, this papers establishes a two-stage stochastic programming (TSSP) framework integrating the activity chains to model activity-level mobility flows using LBSN data. It is worth mentioning that, stochastic programming has been applied to optimize the allocation of traffic sensors considering the uncertainty in the path flow distribution (*11*), and the OD reconstruction problem based on traffic counts (*12*). However, to the best of our knowledge, this is the first effort to apply it to model the dynamic OD estimation problem based on LBSN data.

In this study, we assume that similar check-in patterns are generated by the same OD flow pattern, and these check-in patterns are treated as scenarios in the stochastic programming framework. The first stage minimizes the errors introduced by the inter-zone flows (refer to OD flows hereafter) alongside the expected errors of the check-in patterns. The second stage is to minimize the errors produced by each check-in pattern scenario separately. Note that, a scenario is defined as a realization of the second stage problem state, i.e., the check-in patterns. Finally, the proposed two-stage stochastic programming model is addressed by the generalized Benders decomposition (GBD) algorithm. The idea is to construct a master problem and a series of subproblems (one per scenario) with respect to the first stage and second stage decision variables, respectively. These problem are then optimized alternately and iteratively until the global optimum is found (*11*).

In the remainder of this paper, we first briefly introduce the LBSN check-in data. Then, the mathematical model of the proposed OD estimator based on LBSN data is constructed, followed with the solution algorithm – generalized Benders decomposition. Later on, case studies are elaborated and model performance is evaluated. Finally, we draw some conclusions and suggest future directions for research.

LBSN CHECK-IN DATA DESCRIPTION

This section provides a brief introduction on the characteristics of LBSN check-in data and relevant concepts. An LBSN check-in event is automatically recorded when a user posts with geo-location information or visits a venue (a point-of-interest). Each check-in is described by a user ID, a venue ID, and the time of the check-in. In this regard, we can treat venues as detectors of such events, while users are the objects or counts being detected. Overall, venues and users participate in the services actively as venues can interact with customers in a creative and convenient manner and customers can get awarded (e.g., discounts or "badges") from the social networking sites. Compared to the conventional household surveys, such check-in data can be collected at a very low cost with much higher frequency, and compared to the traffic measurements, detectors (i.e., venues) of check-in events are “deployed” much denser over the urban area.

Combining with the pre-registered location and category information of venues, the check-in data has become a carrier of activity-oriented urban mobility patterns and can thus be used to model the urban travel demand after appropriate aggregation. Normally, venue-side data and user-side data are distinguished in the site server (*3*). Venue-side data contains the check-in statistics with respect to the venue, while user-side data preserves the check-in history of the user. Consequently, one can aggregate the venue-side check-in data based on the categorical hierarchy adopted by the site to model the activity-based mobility flows. Likewise, the activity chains of individuals can be extracted from user-side data. Inspired by this basic idea, in next section we develop a mathematical model for OD estimation using LBSN check-in data which integrates the aggregated check-in patterns of venue-side data and the activity chains extracted from user-side data.

METHODOLOGIES

Given that the OD patterns do not change dramatically within a short period without any disruptive events, we do a plausible assumption that similar check-in patterns at a specific time interval in different days during the reference period are generated by the same OD pattern. Similarly, it is also reasonable to say that OD flows generate when people travel for various activities across different regions within the network. In other words, OD flows are aggregated results of activity flows. Based on the said basic assumption and conceptual analysis, we developed an activity-oriented OD estimator in this section, leveraging the two-stage stochastic programming framework.

In particular, the OD estimator is built upon the graph model shown in Figure 1. For convenience, we define an *activity node* as an aggregating representative for a specific type of activities, e.g., “Food”. An *activity flow* is then the movements of people between two types of activities. For each traffic analysis zone (TAZ), we define a virtual *source* and a virtual *sink* to: (i) “memorize” the sum of in- and out-flows; (ii) counteract the noise in the check-in data collection; (iii) bridge the first-stage and the second-stage model decisions. Noteworthy, for a specific TAZ both the source node and the sink node are connecting to all activity nodes in the TAZ. The proposed approach is to optimize the OD pattern in the first-stage, fulfilling the specific constraints on OD flows and the constraints imposed by the expected cost from the second-stage problem. In the second stage, the check-in pattern will be optimized conditional on a specific problem state and OD pattern. Clearly, the optimal OD pattern and check-in pattern are interdependent.

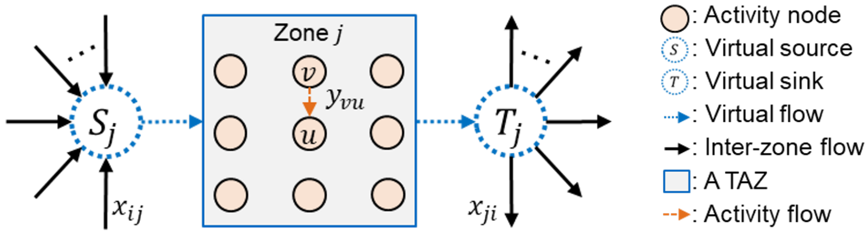


FIGURE 1: Graphical illustration of the model.

Two-Stage Stochastic Programming Model

The check-in patterns are the scenarios considered in the second stage, incurring by a common OD pattern. We formulate the generic two-stage stochastic programming model for OD estimation using LBSN check-in data as follows:

$$\min_{\mathbf{x}} \quad f_1\left(\mathbf{x},\mathbf{x}^{(p)}\right) + \kappa f_c\left(\Phi(\mathbf{x}),\Phi_c(\mathbf{c})\right) + \omega \mathbb{E}_{\xi}\left[Q(\mathbf{x},\xi)\right] \tag{1}$$

$$\text{s.t.} \quad \underline{\varepsilon}_b x_{ij}^{(p)} \leqslant x_{ij} \leqslant \overline{\varepsilon}_b x_{ij}^{(p)} \qquad \forall i,j \in \mathbb{Z} \tag{2}$$

where $\mathbf{x}$ is the decision variable of the first-stage problem, i.e., the vector of OD flows, $\mathbf{x}^{(p)}$ is the given prior OD flows. $f_1(\cdot)$ is the function measuring the difference between the estimated posterior OD flows and the prior OD flows. Similar to traffic measurement based OD estimators, the idea of including $f_1(\cdot)$ in the objective function is to help restrict the search space of the posterior OD flows. $\Phi(\mathbf{x})$ is the vector of out-flows of zones which is obtained by aggregating the estimated OD flows correspondingly, $\Phi_c(\mathbf{c})$ is the given out-flows estimated with the observed check-in statistics $\mathbf{c}$. Note that $f_c(\cdot)$ measures the distance between the modeled and the measured out-flows. The addition of $f_c(\cdot)$ is inspired by the linear relationship observed from ten-month empirical LBSN check-in dataset. Figure 2 compares the observed out-flows and the out-flows estimated by a simple linear regression model based on the number of check-ins, i.e., $\Phi_c(\mathbf{c}) = \hat{\theta}^T \mathbf{c}$, where $\hat{\theta} = (C^T C)^{-1} C^T \Phi_0$, C is the matrix of the number of check-ins aggregated by TAZs, and Φ_0 is the vector of observed out-flows. This out-flows estimator is an input to the proposed OD estimator and will be adopted in the following experiments.

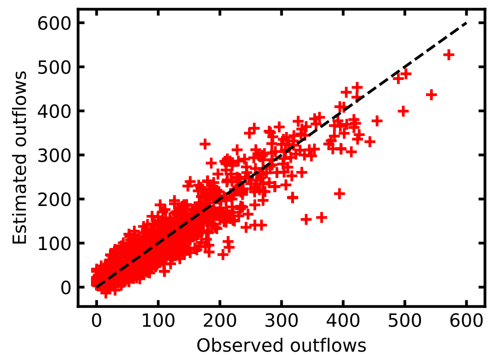


FIGURE 2: Linear relationship between out-flows and the number of check-ins.

While $f_1(\cdot)$ force the posterior OD flows to follow a similar OD pattern as the prior, $f_c(\cdot)$ adjust the OD demand level based on check-in observations. κ is a weight factor that quantifies the relative reliability of the prior OD estimate and the prior out-flow estimate. $\mathbb{Z}$ is the set of TAZs within the study area, and x_{ij} is the OD flow from TAZ i to j. Equation (2) represents the bound constraints on the OD flows, and bounds are defined as a multiple of the prior OD estimate. $\underline{\varepsilon}_b$ (< 1) and $\overline{\varepsilon}_b (> 1)$ are threshold parameters.

Two-stage stochastic programming framework provides an opportunity for further restricting the search space of the OD flows. This is achieved by introducing a batch of check-in pattern scenarios of the second-stage problem state, as expressed by the third term of Equation (1). $\mathbb{E}_\xi$ calculates the expectation with respect to a random vector ξ, defined on the probability space $(\Omega, \mathscr{F}, \mathbb{P})$, with Ω being the sample space, $\mathscr{F}$ being the event space, and $\mathbb{P}$ being a probability distribution defined on $\mathscr{F}$. ξ is a random variable to describe the problem state at the second stage. $\mathbb{E}_\xi[Q(x,\xi)]$ is also called the recourse function. ω is a weight factor that quantifies the trade-off between the optimization of OD flow pattern and check-in patterns. $Q(\mathbf{x},\xi)$ is the optimal value of the second stage problem given by

$$Q(\mathbf{x},\xi) = \min_{\mathbf{y}} \quad \sum_{z\in\mathbb{Z}} f_2\left(\Delta_z(\mathbf{y}_z), \hat{\Delta}_z(\xi)\right) \tag{3}$$

$$\text{s.t.} \quad \sum_{u\in\{\mathbb{V}_z - v\}} y_{vu,z} - \left(\sum_{u\in\{\mathbb{V}_z - v\}} y_{uv,z} + q_{v,z}^{\tau-1}(\xi)\right) \leqslant 0 \qquad \forall v \in \mathbb{V}_z, \forall z \in \mathbb{Z} \tag{4}$$

$$(1-\underline{\varepsilon}_a)\rho_{vu} q_{v,z}^{\tau-1}(\xi) \leqslant y_{vu,z} \leqslant (1+\overline{\varepsilon}_a)\rho_{vu} q_{v,z}^{\tau-1}(\xi) \qquad \forall v,u \in \mathbb{V}_z, \forall z \in \mathbb{Z} \tag{5}$$

$$y_{vu,z} \geqslant 0 \qquad \forall v,u \in \mathbb{V}_z, \forall z \in \mathbb{Z} \tag{6}$$

$$(1-\underline{\varepsilon}_s) \sum_{m\in\{\mathbb{Z}-z\}} x_{mz} \leqslant \sum_{v\in\mathbb{V}_z} y_{sv,z} \leqslant (1+\overline{\varepsilon}_s) \sum_{m\in\{\mathbb{Z}-z\}} x_{mz} \qquad \forall z \in \mathbb{Z} \tag{7}$$

$$(1-\underline{\varepsilon}_t) \sum_{m\in\{\mathbb{Z}-z\}} x_{zm} \leqslant \sum_{v\in\mathbb{V}_z} y_{vt,z} \leqslant (1+\overline{\varepsilon}_t) \sum_{m\in\{\mathbb{Z}-z\}} x_{zm} \qquad \forall z \in \mathbb{Z} \tag{8}$$

where $\hat{\Delta}_z(\xi) = \mathbf{q}_z^{\tau-1}(\xi) - \mathbf{q}_z^{\tau}(\xi)$ is the "check-in pattern" of TAZ z in scenario ξ with $\mathbf{q}_z^{\tau}$ being the vector of the number of check-ins at different activity nodes in time interval τ, $\Delta_z(\mathbf{y}_z)$ is the estimated check-in pattern derived from the optimized activity flows $\mathbf{y}_z$. $f_2(\cdot)$ is a goodness-of-fit function measuring the distance between the observed and estimated check-in patterns. $\mathbb{V}_z$ is the set of activity nodes in TAZ z. In practice, only the main venue categories in the TAZ will be selected for the sake of: (i) reducing the noise in the statistics caused by insufficient venues of a specific category; (ii) distinguishing different TAZs with respect to the land-use functionality and characteristics.

Equation (4) (denominated as inventory constraints) expresses that for a specific activity node v the sum of leaving flows cannot be greater than the sum of the coming flows and the number of check-ins recorded at the previous interval. Equation (5) (denominated as activity share constraints) incorporates the activity chain information extracted from user-side data into the model, which is used to restrict the search space of activity flows and prevent the optimization from overfitting issue to some extent. ρ_{vu} is the activity share of u in the flow out from v, which can be estimated from the historical check-in data. Note, the activity share can be aggregated at either the network level or TAZ level. All activity flows should be non-negative as expressed by Equation (6).

Moreover, recall that we define a source node and a sink node for each zone in the graph model. Thus, it is natural to have the in- and out-flow balance constraints on the source and sink nodes, which connect the decision variables of the first- and second-stage problems. However, due to the randomness and incompleteness of the activity information, we allow a subtle deviation in both constraints. The in-flow balance constraints are then given by Equation (7), representing that for a specific TAZ the sum of activity flows from the source node should not deviate too much from the sum of inter-zone flows to that TAZ. Similarly, the out-flow balance constraints are given by Equation (8), indicating that for a specific TAZ the sum of activity flows to the sink node should not deviate significantly from the sum of inter-zone flows from that TAZ.

$\underline{\varepsilon}_a$, $\overline{\varepsilon}_a$, $\underline{\varepsilon}_s$, $\overline{\varepsilon}_s$, $\underline{\varepsilon}_t$ and $\overline{\varepsilon}_t$ are predefined threshold parameters in the range (0,1). We note that the optimal activity flows $\mathbf{y}^*$ depends on the first-stage OD pattern $\mathbf{x}$ and the second-stage problem state ξ.

By comparison, the proposed LBSN data based OD estimator resembles the generic traffic measurement based OD estimator to a certain extent: (i) the proposed model also relies on a prior OD flow estimate; (ii) $f_c(\Phi(x), \Phi_c)$ in the objective function of the first-stage problem (Equation (1)) and the activity share constraints in Equation (5) play a similar role as the traffic assignment method; (iii) both models need to handle the stochasticity of the observed data. The difference between the two method lies in that LBSN data are collected when the travel is finished but traffic measurements are collected during the travel. In other words, LBSN contains the end-to-end information, while traffic measurements reflects the situation between ends. As a result, the application of LBSN data based OD estimator usually demands no network structure, but needs the information of activity preference of travelers. More importantly, different from most traffic measurement based OD estimators in the existing literature, the proposed model can be used for dynamic OD estimation by only using the LBSN data for estimated check-in patterns without the need of otherwise running computationally expensive dynamic traffic simulation to generate simulated traffic measurements. This puts the proposed methodology on a significant computational advantage against most dynamic OD estimation approaches.

Sampling and Sample Average Approximation

Apparently, the proposed model is very complicated and non-convex as the expectation $\mathbb{E}_\xi$ is usually an integral of a complex function. Accordingly, in practice, we often need to assume ξ has a finite number of possible realizations with a known probability distribution, such that we can estimate $\mathbb{E}_\xi$ by:

$$\mathbb{E}_\xi\left[Q(\mathbf{x},\xi)\right] = \sum_{n}^{N} p_n f_2\left(\Delta_z(\mathbf{y}_z), \hat{\Delta}_z(\xi_n)\right) \tag{9}$$

where N is the total number of realizations. Applying some sampling techniques and sample average approximation (SAA) method, the expectation can then be approximated by:

$$\mathbb{E}_\xi\left[Q(\mathbf{x},\xi)\right] \approx \frac{1}{N_s}\sum_{n}^{N_s} f_2\left(\Delta_z(\mathbf{y}_z), \hat{\Delta}_z(\xi_n)\right) \tag{10}$$

where N_s is the total number of scenario samples selected. In this paper, we apply the k-nearest neighbors algorithm (k-NN) to search for similar check-in patterns in the historical data to compose the set of check-in pattern scenarios of the OD pattern, in which each activity node pair is regarded as one dimension of the observation.

Generalized Benders Decomposition Algorithm

The generalized benders decomposition algorithm (GBD) was first proposed in Geoffrion (*13*) for addressing the mathematical programming problems with complicating variables (i.e., variables that if fixed to given values render a simple or decomposable problem). Obviously, in two-stage stochastic programming, the first-stage decision variables are the complicating variables of the problem. The idea behind GBD is of decomposing the original problem into a master problem and a series of subproblems (one per scenario). In the master problem, the first-stage decisions (i.e., OD flows, $\mathbf{x}$) are optimized. In the subproblems, the second-stage decisions (i.e., activity flows, $\mathbf{y}$) are optimized. They are solved iteratively until convergence. At a specific iteration k, the subproblems are solved first separately resulting in the optimum $\mathbf{y}^k(\xi)$ given $\mathbf{x}^{k-1}$ and scenario ξ. An optimality cut (or feasibility cut) is generated based on the dual solutions of subproblems (or feasibility problem), which is added to the master problem as a new constraint. Given all cut constraints created through a pass-back mechanism from subproblems in previous iterations, the master problem is solved with respect to $\mathbf{x}$ resulting in $\mathbf{x}^k$. Note, these cuts gradually shrink the feasible space of the complicating variable.

To describe the algorithm, we sequentially provide the formulations of the subproblem (SP), the feasibility problem (FP), and the master problem (MP). At the k-th iteration, for a given scenario ξ_n and $\mathbf{x}^{k-1}$, the SP is formulated as follows:

$$\begin{aligned}
\min_{\mathbf{y}} \quad & \sum_{z\in\mathbb{Z}} f_2\left(\Delta_z(\mathbf{y}_z), \hat{\Delta}_z(\xi_n)\right) && (11)\\
\text{s.t.} \quad & \text{Constraints (4)-(8)} && (12)\\
& \mathbf{x} = \mathbf{x}^{k-1}: \quad \lambda_n^k && (13)
\end{aligned}$$

The solution of the SP provides values for the activity flows $\mathbf{y}^k$ in different scenarios, as well as the corresponding optimal Lagrange multipliers vector associated with Constraints (13), i.e., the optimal dual variables vector, λ^k. Note, we can have N_s SPs solved in parallel. If the SP is feasible, the Lagrangian function can be written as:

$$\mathscr{L}_o(\mathbf{x}, \mathbf{y}^k(\xi_n), \lambda_n^k) = \sum_{z\in\mathbb{Z}} f_2\left(\Delta_z(\mathbf{y}_z^k(\xi_n)), \hat{\Delta}_z(\xi_n)\right) + (\lambda_n^k)^T(\mathbf{x} - \mathbf{x}^{k-1}) \tag{14}$$

However, if the SP is infeasible, the following FP will be solved.

$$\begin{aligned}
\min_{\mathbf{y},\eta} \quad & \eta && (15)\\
\text{s.t.} \quad & \text{Constraints (4)-(8)} && (16)\\
& \eta \geqslant 0 && (17)\\
& \mathbf{x} - \mathbf{x}^{k-1} \leqslant \eta: \quad \mu_n^k && (18)
\end{aligned}$$

Similarly, we can get the Lagrangian multiplier vector μ_n^k for Constraints (18). The Lagrangian function of the FP is given by:

$$\mathscr{L}_f(\mathbf{x}, \mu_n^k) = (\mu_n^k)^T(\mathbf{x} - \mathbf{x}^{k-1} - \eta) \tag{19}$$

Then, the MP is formulated as follows:

$$\begin{aligned}
\min_{\mathbf{x},\alpha} \quad & f_1\left(\mathbf{x}, \mathbf{x}^{(p)}\right) + \kappa f_c\left(\Phi(\mathbf{x}), \Phi_c(\mathbf{c})\right) + \alpha && && (20)\\
\text{s.t.} \quad & \underline{\varepsilon}_b x_{ij}^{(p)} \leqslant x_{ij} \leqslant \overline{\varepsilon}_b x_{ij}^{(p)} && \forall i, j \in \mathbb{Z} && (21)\\
& \omega/N_s \sum_{n=1}^{N_s} \mathscr{L}_o(\mathbf{x}, \mathbf{y}^t(\xi_n), \lambda_n^t) \leqslant \alpha && \forall t \in \mathbb{I}_o && (22)\\
& \mathscr{L}_f(\mathbf{x}, \mu_l^t) \leqslant 0 && \forall l \in \mathbb{S}_f^t, \forall t \in \mathbb{I}_f && (23)
\end{aligned}$$

where $\mathbb{I}_o$ is the set of indices of the iterations at which all SPs are feasible, $\mathbb{I}_f$ is the set of indices of the iterations at which at least one of the SPs is infeasible, and $\mathbb{S}_f^t$ is the set of scenarios whose associated SPs are infeasible at iteration t. Constraints (22) are denominated as optimality cuts, while Constraints (23) are feasibility cuts. From MP, we can get the values of the first-stage decision variables $\mathbf{x}^k$.

If the original objective function is convex on the complicating variable, GBD can guarantee the strong optimality condition, i.e., the optimal solution from the decomposed problems is equivalent to the original problem. For convenience, further details on the procedure of the GBD algorithm are presented in Algorithm 1.

Algorithm 1 Generalized Benders decomposition algorithm for OD estimation

Initialize the OD flows $\mathbf{x}_0$.
Initialize the iteration index $k = 1$, the complicating variables $\mathbf{x}^k = \mathbf{x}_0$, error tolerance ε, the maximum number of iterations M.
Set the lower bound of the objective function $\underline{z}^k = 0$, and the upper bound $\overline{z}^k = \infty$.
while $|\overline{z}^k - \underline{z}^k|/|\underline{z}^k| \geqslant \varepsilon$ and $k < M$ **do**
    Set $k := k+1$.
    Solve the subproblems by fixing $\mathbf{x}$ as $\mathbf{x}^{k-1}$.
    if all subproblems are feasible **then**
        Obtain solution $\mathbf{y}^k$ and the dual variables of those constraints that fix the complicating variables to given values λ^k.
        Calculate $z = f_1(\mathbf{x}^{k-1}, \mathbf{x}^{(p)}) + \kappa f_c(\Phi(\mathbf{x}^{k-1}), \Phi_c(\mathbf{c})) + \omega/N_s \sum_n^{N_s} \sum_z f_2\left(\Delta_z(\mathbf{y}_z), \hat{\Delta}_z(\xi_n)\right)$.
        Update the upper bound $\overline{z}^k = \min\{\overline{z}^{k-1}, z\}$.
        Set $\mathbb{I}_o := \mathbb{I}_o \cup \{k\}$.
    else
        Solve the feasibility problems associated with the infeasible subproblems.
        Obtain solution $\mathbf{y}^k$, dual variable vector μ^k and the set of infeasible subproblems $\mathbb{S}_f^k$.
        Set $\mathbb{I}_f := \mathbb{I}_f \cup \{k\}$.
    end if
    Add the new optimality cut (or feasibility cuts) to the master problem.
    Solve the master problem to get $\mathbf{x}^k$ and α^k.
    Update the lower bound $\underline{z}^{(k)} = f_1(\mathbf{x}^k, \mathbf{x}^{(p)}) + \kappa f_c(\Phi(\mathbf{x}^k), \Phi_c(\mathbf{c})) + \alpha^k$.
end while

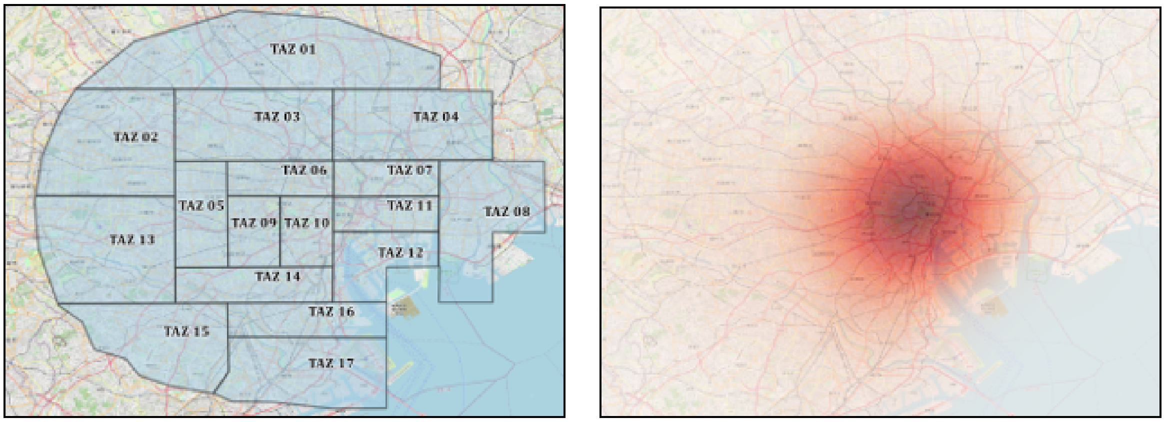
EXPERIMENTAL DESIGN

In this section, we verify the proposed OD estimator using the Foursquare check-in data. Foursquare was launched in 2009 and has provided the leading LBSN service for more than a decade. As of 2016, it has more than 60 million registered users, with over 50 million monthly active. At the same time, more than 95 million venues from over 190 countries or regions are registered on the site. Their real-world images and consumer reviews are being updated constantly. It indicates that the Foursquare data has a broad spatial coverage and therefore can somewhat capture the human behavior in urban areas.

Case study setup

The Foursquare check-in data (*14*) of Tokyo city, Japan, from April 2012 to February 2013 are used in the following experiments. Figure 3a shows the map of the study area and the delineation of TAZs. It is clear that the study area (1,302 km^2) is divided into 17 TAZs. Figure 3b exhibits a heatmap of 10,000 check-in records randomly sampled from the entire dataset, which contains

57,3703 records. The heatmap has a clear center and the color intensity gradually fades from the center outward. We note that TAZs are devised based on the density of check-ins for the sake of statistical significance, i.e., denser area has more TAZs.



(a) TAZs

(b) Heatmap of check-ins

FIGURE 3: Study area: City of Tokyo.

Due to the lack of venue-side data, we aggregate user-side data hourly for each parent venue category[1], each TAZ, and each day to reconstruct the venue-side dataset. Categories with fewer than five check-ins are not further defined as an activity node of the TAZ. This can help identify the functionality of TAZs in land use and the generator of their attractiveness over time. For instance, the TAZ that has a huge number of "College & University" check-ins but a negligible quantity of "Professional & Other Places" check-ins is more likely an area including higher-educational institutions. Futhermore, we apply the moving average (seven days) technique to cancel the randomness of check-in behavior. In terms of the user-side dataset, for a specific time interval, we first extract the activity chain of each user. An activity share matrix can then be derived by counting the number of transfers between every two activities followed with normalization.

Algorithm setup

In the following experiments, we set the threshold parameters, $\{\underline{\varepsilon}_s, \overline{\varepsilon}_s, \underline{\varepsilon}_t, \overline{\varepsilon}_t, \underline{\varepsilon}_a, \overline{\varepsilon}_a\}$, as 0.2. The bound constraint parameters, $\{\underline{\varepsilon}_b, \overline{\varepsilon}_b\}$, are 0.2 and 5, respectively. We set the weight factors, κ and ω, as 1. The number of second stage realizations N_s is set to 5. Regarding the GBD algorithm, the convergence threshold is $\varepsilon = 0.05$. Without loss of generality, we apply the generalized linear squares (GLS) estimator in the goodness-of-fit functions, $f_1(\cdot), f_c(\cdot)$ and $f_2(\cdot)$, resulting in a convex optimization problem with complicating variables. In this case, the GBD algorithm can guarantee a global optimal solution as the original problem. Mathematically, $f_1(\cdot), f_c(\cdot)$ and $f_2(\cdot)$ are given as follows:

$$f_1\left(\mathbf{x}, \mathbf{x}^{(p)}\right) = \left(\mathbf{x} - \mathbf{x}^{(p)}\right)^T \Lambda_1 \left(\mathbf{x} - \mathbf{x}^{(p)}\right) \tag{24}$$

[1]Foursquare leverages its own proprietary taxonomy of more than 1000 categories. According to the hierarchical taxonomy of categories (version 2012), ten parent categories are defined, including: Arts & Entertainment, College & University, Event, Food, Nightlife Spot, Outdoors & Recreation, Professional & Other Places, Residence, Shops & Service, Travel & Transport.

$$f_c\left(\Phi(\mathbf{x}),\Phi_c(\mathbf{c})\right) = \left(\Phi(\mathbf{x})-\Phi_c(\mathbf{c})\right)^T \Lambda_c\left(\Phi(\mathbf{x})-\Phi_c(\mathbf{c})\right) \tag{25}$$

$$f_2\left(\Delta_z(\mathbf{y}_z),\hat{\Delta}_z(\xi_n)\right) = \left(\Delta_z(\mathbf{y}_z)-\hat{\Delta}_z(\xi_n)\right)^T \Lambda_2\left(\Delta_z(\mathbf{y}_z)-\hat{\Delta}_z(\xi_n)\right) \tag{26}$$

where Λ_1, Λ_c and Λ_2 are the dispersion matrix of the prior OD estimate, of the out-flows distribution, and of the check-in pattern, respectively. For simplicity, we set $\Lambda_1 = \Lambda_c = \Lambda_2 = \text{diag}(\mathbf{1})$.

Demand scenario setup

To conduct the experiments, we chose the morning peak (7 am - 10 am) of February 1st, 2013, divided in three estimation time intervals each for one hour. The entire check-in dataset is used for estimating the relationship between the number of check-ins and the out-flows, i.e., $\hat{\theta}$, as aforementioned. Further to generate the demand estimation scenarios, Antoniou et al. (*5*) points out that the quality of the prior OD estimate, in terms of both demand level and patterns, is a key element affecting the performance of the OD estimator. Following the suggestions therein, we perturb the true OD flows to derive the historical OD flow estimates to be provided as inputs for the OD estimator. Due to space limitations, here we only test the performance of the proposed approach in low-demand scenarios, i.e., the prior OD estimate is out-of-date and is lower than the true demand level. More specifically, we create the prior OD estimate using the following equation:

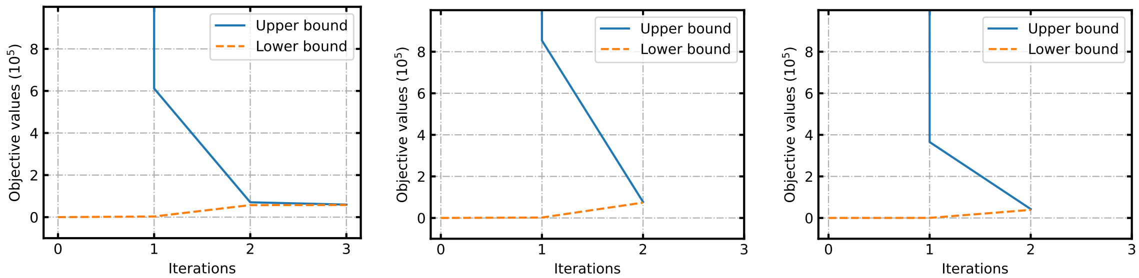
$$\mathbf{x}^{(p)} = (0.7+0.3\delta)\mathbf{x} \quad \delta \sim \mathcal{N}(0,1/3) \tag{27}$$

RESULTS

In this section, we first analyze the convergence performance of the GBD algorithm. Then, the fit of the estimated OD flows, to the true OD demand, to the check-in pattern, and to the activity share is presented. Finally, we show that the LBSN OD matrix can be easily scaled up to approximate the network OD matrix, illustrating the potential of the proposed OD estimator for practical applications.

Algorithm Convergence Analysis

Figure 4 depicts the convergence results for estimating the OD matrices of the three experiment intervals. Since the initial upper bound is infinity, it is not visible in the figure. Recalled that the upper bound is updated by solving the subproblems, while the lower bound is updated by solving the relaxed master problem. As expected, the algorithm converges within only a few iterations in all three experiments, due to the convexity of the problem (GLS estimator). At each iteration, the algorithm needs to solve the relaxed master problem once, and all the N_s subproblems in parallel. Note, the relaxed master problem can be solved efficiently (in seconds) as it has limited number of constraints and all constraints share the same format. We also found that subproblems are always feasible if $\mathbf{x}_0$ is feasible, which means all Benders type cuts are optimality cuts, and thus no feasibility problems needed to solve and no feasibility cuts are added to the master problem. In consequence, the three experiments are solved in rather cheap computational efforts, indicating that the proposed modeling framework has the potential for estimating the dynamic OD matrices for large scale networks.



(a) 7 am - 8 am (b) 8 am - 9 am (c) 9 am - 10 am

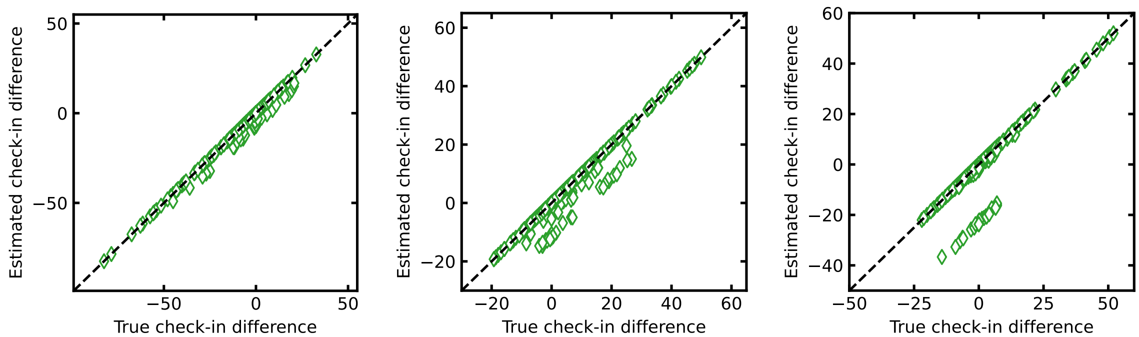
FIGURE 4: Convergence performance of the GBD algorithm.

Estimation Quality Evaluation

Figure 5 illustrates the quality of estimation with respect to the check-in patterns by comparing the empirical and estimated check-in pattern using 45° plots. Recall that check-in pattern is defined as the map of the difference of check-in counts between successive time stamps. We can see that all points are aligned closer to the "$y = x$" line at interval 7 am - 8 am, confirming the capability of the model in recreating the check-in pattern. However, in intervals 8 am - 9 am and 9 am - 10 am, though most points are also located near the 45° line, some deviate relatively further from the line. It is caused by the usage of k-NN in scenario selection as described in the previous section. k-NN would lead to biased results under the situation of limited candidate set. As a result, OD estimates that are promising to some check-in scenarios accidentally incur a biased estimate to some activity nodes in the others. The problem could be eliminated by incorporating the model with better sampling methods. Moreover, we can also see that some venues observe large negative check-in difference in the interval 7 am - 8 am. These venues may represent the residence places given that people are more likely to leave their home to work at this time. In all three intervals, we can find that most points are located in the range $[-20, 20]$, which represent the regular movements between different activities.

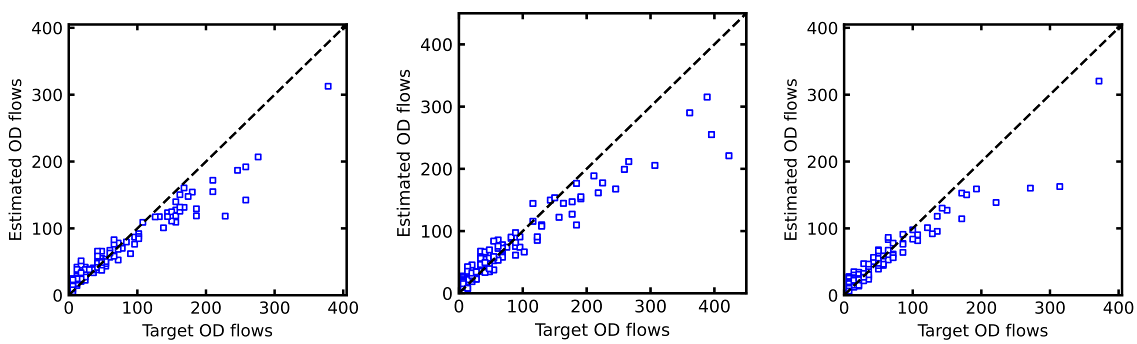
Similarly, Figure 6 visualizes the quality of the estimated OD flows by comparing it with the target OD flows using 45° plots. Overall, the model reaches to an acceptable estimate with a slight underestimation in the high demand OD pairs. We note that the prior OD estimate is also underestimated as 70% of the target values in average. Our model can somehow improve the situation attributed to the inclusion of $f_c(\cdot)$ and a batch of check-in pattern scenarios.

Further, Figure 7 compares the theoretic activity shares and the estimated activity shares. The difference has been restricted by the activity share constraints expressed in Equation (5). Due to the large number of points, we add the heatmap effect in the figure to represent the density of points. Brighter colors mean greater density, vice versa. Overall, there are more points in the range of smaller values. Since the constraints are defined based on percentage values, it is plausible to see the points with larger values are more scattered. In addition, we also note that more active movements (larger activity shares present) can be observed in 8 am - 9 am and 9 am - 10 am. Hence, together with the label of these activity flows, Figure 7 can provide useful auxiliary information for dynamic traffic management, and help venues design working schedules and "production" plans.



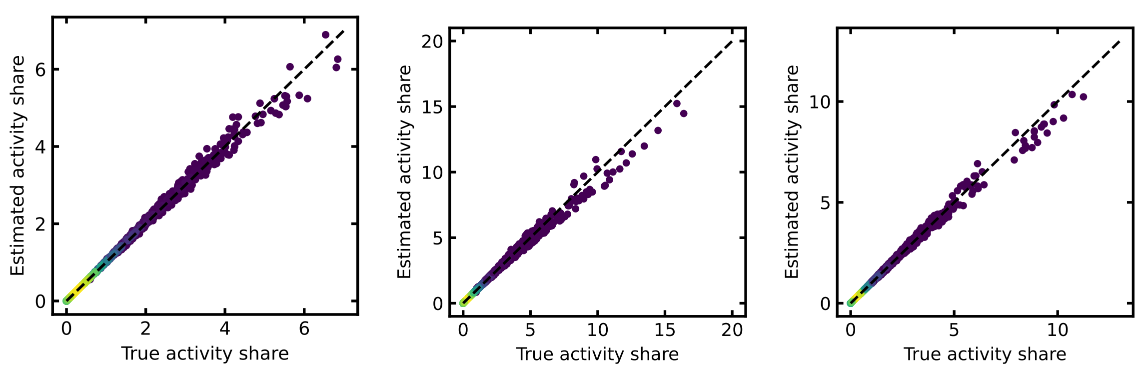
(a) 7 am - 8 am (b) 8 am - 9 am (c) 9 am - 10 am

FIGURE 5: Comparison of true and estimated check-in patterns.



(a) 7 am - 8 am (b) 8 am - 9 am (c) 9 am - 10 am

FIGURE 6: Comparison of target and estimated OD flows.



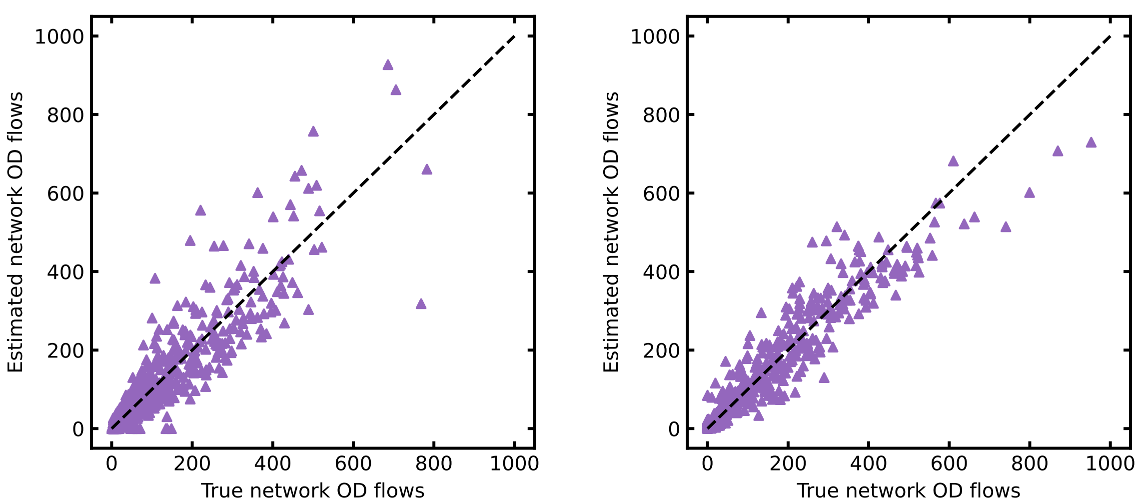
(a) 7 am - 8 am (b) 8 am - 9 am (c) 9 am - 10 am

FIGURE 7: Comparison of true and estimated activity share.

Scaling Towards Network OD Flows

The OD flows compared in the previous sections only represent a partial set of the overall network OD flows that are observed in the LBSN check-in dataset (refer to LBSN OD matrix/flows hereafter). Whilst, we usually need the actual network OD matrix in real practical applications instead. It is a critical input to the traffic simulation models for evaluating traffic management and policy measures (*5*). However, we argue that due to the random nature of check-in behaviors the LBSN OD matrix follows the same structural pattern as the network OD matrix, except for the deviation in the magnitude of demand level. Therefore, Figure 8 compares the network OD flows and the magnified LBSN OD flows using a similar method as in Figure 2. More specifically, the "true" network OD flows here are the average of TomTom OD flows of all Wednesdays in January 2021. The estimated network OD flows are obtained by scaling up the observed LBSN OD flows using a common scaling matrix for morning time and another matrix for afternoon time. Despite a rough estimation, the estimated ODs has a close pattern to the true ODs. It is reasonable to say that the Scatteredness of points is the joint result of the combination of the long distance in time (that the two datasets are collected) and the bias in data collection methods.

TomTom is a Dutch corporation launched in 1991 that specializes in the production of car navigation systems of all types. Many enterprises use TomTom's positioning technologies such as Microsoft and Uber, due to their high precision. The TomTom data are collected from the probe vehicles which are equipped with TomTom's positioning devices. Given the high penetration rate of TomTom positioning devices, we believe that TomTom data can capture the real traffic state to a large extent. Therefore, Figure 8 confirms that the LBSN OD flows estimated by the proposed model can approximate the real network OD flows after appropriate scaling. Theoretically, the scaling matrix can be either used after the LBSN OD estimation or directly integrated into the estimation framework. Also note that, exploration of more suitable scaling method which can incorporate demand information such as trip length frequency distributions to improve the network OD estimation are part of our future work.



(a) Morning (7 am - 13 pm) (b) Afternoon (13 pm - 19pm)

FIGURE 8: Comparison of true and estimated network OD flows.

CONCLUSION

Origin-destination (OD) estimation methods have long relied on the two traditional data sources of household travel surveys and traffic network detection. On the one hand, travel surveys are time-consuming and labor-intensive, restraining their coverage and frequencies, while on the other, the traffic detection infrastructure is expensive to install and maintain, restraining its density and triggering indeterminateness issues for OD estimation methods. Therefore, OD estimation methods that utilize inexpensive and widespread data sources

Considering the stochastic nature of human behaviors and transportation systems, we propose a dynamic OD estimator by utilizing the scenario-based two-stage stochastic programming framework, which integrates the activity chains extracted from LBSN check-in data to model activity-level mobility flows. Given that OD flows are aggregated results of activity flows, the OD matrix can be derived from these activity flows. Within the framework, the first stage model aims to minimize the errors introduced by the inter-zone OD flows alongside the expected errors of the check-in patterns. At the same time, the second stage model attempts to minimize the errors produced by the considered check-in pattern scenarios. To solve the two-stage stochastic programming model, a generalized Benders decomposition is presented, which seeks the optimal solution by solving a relaxed master problem and a series of subproblems iteratively.

To evaluate the approach, we employ the case study of Tokyo city and use the generalized least squares (GLS) estimator to measure the goodness of fit of both check-in and OD patterns. The experiment results show that the convergence of the algorithm can be guaranteed within several steps. Note, since our model is simulation free, the computational efficiency can be significantly improved. More Importantly, the model leads to a good fit for check-in patterns, OD flows, and activity share distributions. Furthermore, we also present the method to scale up the LBSN OD matrix to approximate the real network OD matrix, which can inspire the implementation of the proposed model in practical applications.

Future directions for research would be to integrate appropriate sampling methods (e.g., importance sampling) into the proposed estimation framework for generating significant scenarios instead of simply applying *k*-NN for scenario selection. On the other hand, embedding a suitable scaling method into the model will also be useful.

ACKNOWLEDGEMENTS

This work was supported by the European Interest Group CONCERT-Japan DARUMA project (Grant Number: 01DR21010) funded by the German Federal Ministry of Education and Research (BMBF). This research was also partially funded by the German Research Foundation DFG (TRAMPA Project, Grant 415208373).

REFERENCES

1. Ren, J. and Q. Xie, Efficient OD trip matrix prediction based on tensor decomposition. *Proceedings - 18th IEEE International Conference on Mobile Data Management, MDM 2017*, 2017, pp. 180–185.
2. Xiong, X., K. Ozbay, L. Jin, and C. Feng, Dynamic Origin–Destination Matrix Prediction with Line Graph Neural Networks and Kalman Filter. *Transportation Research Record*, Vol. 2674, No. 8, 2020, pp. 491–503.
3. Yang, F., P. J. Jin, Y. Cheng, J. Zhang, and B. Ran, Origin-destination estimation for non-commuting trips using location-based social networking data. *International Journal of Sustainable Transportation*, Vol. 9, No. 8, 2015, pp. 551–564.
4. Cascetta, E., A. Papola, V. Marzano, F. Simonelli, and I. Vitiello, Quasi-dynamic estimation of o–d flows from traffic counts: Formulation, statistical validation and performance analysis on real data. *Transportation Research Part B: Methodological*, Vol. 55, 2013, pp. 171–187.
5. Antoniou, C., J. Barceló, M. Breen, M. Bullejos, J. Casas, E. Cipriani, B. Ciuffo, T. Djukic, S. Hoogendoorn, V. Marzano, L. Montero, M. Nigro, J. Perarnau, V. Punzo, T. Toledo, and H. van Lint, Towards a generic benchmarking platform for origin–destination flows estimation/updating algorithms: Design, demonstration and validation. *Transportation Research Part C: Emerging Technologies*, Vol. 66, 2016, pp. 79–98.
6. Qurashi, M., Q.-L. Lu, G. Cantelmo, and C. Antoniou, Dynamic demand estimation on large scale networks using Principal Component Analysis: The case of non-existent or irrelevant historical estimates. *Transportation Research Part C: Emerging Technologies*, Vol. 136, 2022, p. 103504.
7. Hu, W. and P. J. Jin, An adaptive hawkes process formulation for estimating time-of-day zonal trip arrivals with location-based social networking check-in data. *Transportation Research Part C: Emerging Technologies*, Vol. 79, 2017, pp. 136–155.
8. Jin, P. J., M. Cebelak, F. Yang, J. Zhang, C. M. Walton, and B. Ran, Location-based social networking data: exploration into use of doubly constrained gravity model for origin–destination estimation. *Transportation Research Record*, Vol. 2430, No. 1, 2014, pp. 72–82.
9. Kheiri, A., F. Karimipour, and M. Forghani, Intra-urban movement flow estimation using location based social networking data. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. 40, No. 1, 2015, p. 781.
10. Cho, Y.-S., G. Ver Steeg, and A. Galstyan, Where and Why Users" Check In". In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2014, Vol. 28.
11. Fu, C., N. Zhu, and S. Ma, A stochastic program approach for path reconstruction oriented sensor location model. *Transportation Research Part B: Methodological*, Vol. 102, 2017, pp. 210–237.
12. Jeong, I.-J. and D. Park, Stochastic programming approach for static origin–destination matrix reconstruction problem. *Computers & Industrial Engineering*, Vol. 157, 2021, p. 107373.
13. Geoffrion, A. M., Generalized benders decomposition. *Journal of optimization theory and applications*, Vol. 10, No. 4, 1972, pp. 237–260.

14. Yang, D., D. Zhang, V. W. Zheng, and Z. Yu, Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 45, No. 1, 2014, pp. 129–142.