

# Time Dependent Origin-destination Estimation from Traffic Count without Prior Information

Hsun-Jung Cho · Yow-Jen Jou · Chien-Lun Lan

Published online: 17 December 2008  
© Springer Science + Business Media, LLC 2008

**Abstract** Existing research works on time-dependent origin-destination (O-D) estimation focus on the surveillance data usually assume the prior information of the O-D matrix (or transition matrix) is known (or at least partially known). In this paper, we relax such assumption by combining Gibbs sampler and Kalman filter in a state space model. A solution algorithm with parallel chain convergence control is proposed and implemented. To enhance its efficiency, a parallel structure is suggested with efficiency and speedup demonstrated using PC-cluster. Two numerical examples (one for Taipei Mass Rapid Transit network and the other for Taiwan Area National Freeway network) are included to show the proposed model could be effective of time-dependent origin-destination estimation.

**Keywords** Time-dependent origin-destination estimation · State space model · Gibbs sampler · Kalman Filter · Parallel computing

## 1 Introduction

This investigation considers a time dependent origin-destination (O-D) estimation involving the state space model, Gibbs sampler, and Kalman Filter. The O-D matrices, representing the number of traffic moving between node (or zone) pairs of the transportation network, are fundamental information of the transportation planning, operation, and control problems. The static O-D matrices are interesting owing to its important role in numerous real world problems, namely, transportation

---

H.-J. Cho (✉) · C.-L. Lan  
Department of Transportation Technology and Management, National Chiao Tung University,  
Hsinchu, Taiwan 300, China  
e-mail: hjcho@cc.nctu.edu.tw

Y.-J. Jou  
Department of Information and Finance Management, National Chiao Tung University, Hsinchu,  
Taiwan 300, China

planning, urban and regional planning; an average zone-to-zone O-D matrix for a whole period of interest is acceptable for this purpose. Regarding the time dependent O-D, it is particularly crucial in the field of transportation operation and management; most advanced traveler information system (ATIS) and advanced traffic management system (ATMS) require time-dependent O-D information as an input for generating traffic evolution over the interested network (Bernstein et al. 1996, 2001; Chang and Wu 1994; Jou and Hwang 2002; Lee et al. 2001). O-D information is not only important in the traffic network but also crucial for the IP network; some researchers had identified the issues faced in inferring the transportation network of an IP network (Benamer and Roberts 2004). Because of the importance of the O-D estimation, many researches had focused on this problem for the past three decades. Customary O-D estimations can be classified into three major methodologies including (1) extensive data survey, i.e., home-based survey, road-side survey, license plate recognizing, and so on; (2) trip distribution models, i.e., gravity model, assignment-based model; and (3) estimation of the O-D matrices by non-assignment-based model using traffic counts of the link derived from the surveillance system.

The first approach, an extensive panel survey, would yields accurate results when conducted carefully. This approach often involves home-based survey, road-side survey, license plate recognizing, and so on; all of them require a large amount of resources and are not possible to undertake frequently. This survey method is time-consuming and not able to give real-time results of O-D matrices; hence, it is not suitable for traffic operation problems, and other real-time applications. This survey data can be treated as priori information about the network and be updated during the estimation period (Doblas and Benitez 2005). By using new technologies, such as Automatic Vehicle Identification (AVI), to sample the O-D matrix can lower the cost of survey process (Dixon and Rilett 2005). But in the real world, the installation density of such AVI facilities on the network is still far from practical. Jou had proposed a statistical approach to estimate the O-D matrix from such incomplete sample, but it still cannot fulfill the need of dynamic O-D (Jou et al. 2006).

The second approach, estimate O-D matrices by trip distribution models. These approaches include gravity model, Tobit model, and dynamic traffic assignment (DTA); among these models, the assignment based models have caught most attentions in the past two decades. These models are based on the assumption that reliable model for describing network flow assignment is available for generating the link flow pattern; entropy maximization, maximum likelihood methods, and generalized least squares methods have been widely used (Van Zuylen and Willumsen 1980; Maher 1983; Cascetta 1984; Bell 1991). A comprehensive review of researches in this approach can be found in Cascetta and Nguyen (1988). Willumsen was the first researcher who extended the entropy concept to multiple time intervals, and Nguyen was the first to address the O-D estimation problem from a network equilibrium approach (in technical report 60 of University of Montreal 1977). Cascetta and et al. extended the statistical estimators of updating O-D matrices, developed by Cascetta and Nguyen (1988), and established two dynamic estimators for approximating the dynamic O-D matrices (Cascetta et al. 1993). More recently, Ashok and Ben-Akiva introduced stochasticity to map the assignment matrix between time-dependent O-D flows and link volumes both in off-line and

real-time applications (Ashok and Ben-Akiva 2002). Lo and Chan further improved the assignment-based model by propose an statistical approach to simultaneous estimate an O-D matrix and link choice proportions from O-D survey data and traffic counts (Lo and Chan 2003). Nie and Zhang proposed a relaxation approach for estimating static O-D matrix that minimizes a distance metric between measured and estimated traffic condition while the condition satisfies user equilibrium (Nie and Zhang 2008). It is worthy to note, for these models developed with this approach to effectively estimate the O-D matrices, there should exist a reliable assignment model and an accurate consecutive interval of O-D matrices. The real challenge of this approach lies on a reliable descriptive dynamic network assignment model remained to be developed by researchers.

The third approach, estimate the O-D matrices by non-assignment models. O-D estimation methods in this category estimate the O-D matrices directly from time-series of network flows. It assumes that each O-D parameter remains approximately constant during several consecutive intervals of estimations. The concept was originally proposed by Cremer and Keller for identifying turning counts from link traffic count, with the assumption of constant link travel time (Cremer and Keller 1981). Most non-assignment based studies are conducted successfully against simple networks with assumption that most entering and leaving flows at network origin/destination nodes are available. Chang and Wu proposed a model employs information from mainline traffic counts, ramp flow measurements, and macroscopic traffic characteristics to construct a set of dynamic equations to describe the O-D distributions and observed flows (Chang and Wu 1994). Chang and Tao further employed the link flow counts and specially-designed dynamic screenline flows to estimate the O-D matrix; the target network is also decomposed into several sub-networks for parallel computing (Chang and Tao 1995). The limitation of this approach lies on the limited traffic surveillance facilities in a real-world network. Most non-assignment-based models require considerably large amounts of link flow information for computing the O-D flow; these models in general are also too computationally intensive for real-time implementations in large-scale networks.

Okutani first introduced state space model into time-dependent O-D estimation with the state vector indicating the unknown O-D flows; prior data is required to identify the parameter matrix (Okutani 1987). In 1993, Ashok and Ben-Akiva introduced another state space model approach with the state vector defined in terms of O-D flow deviations (Ashok and Ben-Akiva 1993). Jou suggested an approach based on state-space modeling with the state vector also indicates the O-D flows, but no prior information of state vectors and transition matrices are needed; while most of the aforementioned studies assume that the transition matrix is known or at least approximately known, which is unrealistic for a real world network (Jou and Hwang 2002). Kalman Filter is an efficient recursive filter that estimates the state of a dynamic system from a series of incomplete and noisy measurements. This character is similar to the behavior of traffic; the link flow counts derived from vehicle detectors are noisy. While the least square method, being widely used in many researches, is mainly used to solve overdetermined systems, which is usually violated in traffic system.

In this paper, Gibbs sampler is combined with Kalman filter to estimate the state vector and transition matrix simultaneously; parallel technique is introduced to the

proposed algorithm and improved the performance. We begin, in Section 2, with the introduction of time-dependent origin destination estimation by state space model with Gibbs sampler. The parallel chain convergence control and parallel implementation is addressed in Section 3, numerical examples are demonstrated in Section 4, and the conclusion is outlined in the last section.

## 2 Time-dependent origin-destination estimation

Existing research works on traffic assignment usually assume the existence of user equilibrium status or pursuit some system optimal situation. However, it is questionable whether such equilibrium state really exists or not. In this study, we relax such assumption by some statistical approaches. The basic assumption in this research is the path-flows in time-series have an unknown relationship, the assumption is similar to Okutani does. While Okutani assumed a time invariant relation between time-series O-D flow exists; and this relationship can be estimated by some prior information. In this study, we do not estimate the relationship by prior information. The relationship is estimated simultaneously with path flows.

For convenience, we use the following notations in this chapter as Table 1:

### 2.1 Time-dependent origin-destination estimation by state space model

In this subsection, only the most basic modeling is introduced. The transition matrix that describe the relationship among path-flows in different time period is assumed to be fixed. The link travel time, which is an important issue, is not considered. The above assumptions are made in this section; while the travel time issue will be considered in the next section.

State space model is introduced to estimate path flows from link traffic counts. The state space model is coupled with two parts: transition equations and observation equations. First, the state equation which assumed that the path flows at time  $t$  can be related to the path flows at time  $t-1$  by the following autoregressive form,

$$x_t = Fx_{t-1} + u_t, \quad t = 1, 2, 3, \dots, n \quad (1)$$

**Table 1** Notations

Notation	Descriptions
$F$	a $p \times p$ path flow transition matrix
$x_t$	a $p \times 1$ network path flows on time $t$
$x_t^{ag}$	a $p(MaxLag+1) \times 1$ matrix composed of series of path flows
$y_t$	a $q \times 1$ link traffic observation vector on time $t$
$H$	a $q \times p$ zero-one matrix which denotes the path-observation incidence matrix
$H_t$	a $q \times p(MaxLag+1)$ zero-one matrix which denotes the path-observation incidence matrix at time $t$
$P$	number of elements in path set $P$
$Q$	number of observations in the target network
$u_t, v_t$	independently and identically distributed Gaussian noise terms
$T$	The transport of a matrix is mark by superscript $T$

where  $x_t$  is the state vector which is unobservable,  $F$  is a random transition matrix,  $u_t \sim N_p(0, \Sigma)$  is independently and identically distributed noise term, where  $N_p$  denotes the  $p$ -dimensional normal distribution,  $\Sigma$  is the corresponding covariance matrix.  $x_t$ , the  $p \times 1$  state vector, is defined to be the path flows belonging to O-D pairs. Next, the observation equation,

$$y_t = Hx_t + v_t, \quad t = 1, 2, 3, \dots, n \tag{2}$$

where  $y_t$  is the  $q \times 1$  observation vector which means there are  $q$  detectors on the network. The number of paths is denoted by  $p$ .  $H$  is a  $q \times p$  zero-one matrix, which denotes the path-observation incidence matrix. The path-observation is pre-determined by generating possible path set and given travel time on each link.  $v_t$  is also a noise term that  $v_t \sim N_q(0, \Gamma)$ .

### 2.2 Time-dependent origin-destination estimation by state space model considering travel time

Travel time is an important issue while facing transportation problems. It is no exception in path flow estimation. We consider travel time effect into the estimation model in this subsection.

The model in this section had been modified from the previous section (Section 2.1) to consider travel time effect. The state transition equation is the same as in previous section, but the observation equation had been modified as follows.

$$x_t = Fx_{t-1} + u_t, \quad t = 1, 2, 3, \dots, n \tag{3}$$

$$y_t = H_t x_t^{lag} + v_t, \quad t = 1, 2, 3, \dots, n \tag{4}$$

The state variable,  $x_t^{lag}$ , is modified here to take the travel time into account. It is a  $p \times (MaxLag+1)$  matrix composed of series of state vectors,

$$x_t^{lag} = \left\{ x_t^T, x_{t-1}^T, x_{t-2}^T, \dots, x_{t-MaxLag}^T \right\}^T$$

Let the element in  $x_t$ , a  $p \times 1$  vector, be represented as  $\begin{bmatrix} x_t(1) \\ x_t(2) \\ \vdots \\ x_t(p) \end{bmatrix}_{p \times 1}$ .  
Then,

$$x_t^T = [x_t(1)x_t(2) \cdots x_t(p)]_{1 \times p},$$

and

$$x_t^{lag} = \left\{ [x_t(1) \cdots x_t(p)] [x_{t-1}(1) \cdots x_{t-1}(p)] \cdots [x_{t-MaxLag+1}(1) \cdots x_{t-MaxLag+1}(p)] [x_{t-MaxLag}(1) \cdots x_{t-MaxLag}(p)] \right\}_{1 \times p(MaxLag+1)}^T$$

The term *MaxLag* denotes the maximum time lag intervals that can be observed on observation sites at time  $t$ . The *path-observation* incidence matrix,  $H_t$ , is a zero-one  $q \times p(MaxLag+1)$  matrix at time  $t$ . If the flow of a path can be observed at detector  $q$ , then the corresponding element in  $H_t$  is one; else, it is zero.

Both  $x$  and  $F$  are unobservable, thus Kalman filter is not suitable to directly estimate and forecast the state vector. Hence, Gibbs sampler is used to tackle the problem of simultaneous estimation of  $F$  and  $x$  by available information.

There are two major elements to be incorporated in the solution method, (1) filtering states by observations, and (2) sampling scheme of transition matrix,  $F$ , and state vector,  $x$ . Since the observations,  $y_t$ , are not used directly in the conditional distribution, the Kalman filter and the Gibbs sampler must be combined. For simplicity, in the following section of Kalman Filter and Gibbs sampler, the time-dependent path-observation incidence matrix,  $H_t$  is denoted as  $H$ . After the estimation of state vector, O-D flows can be calculated by the summation of path flows.

### 2.3 Kalman filter

State-space model in conjunction with Gibbs sampler and Kalman filter had been used in a wide variety of applications in many disciplines. Kalman filter gives an estimator for the linear–quadratic–Gaussian problem, which estimates the instantaneous state of a linear dynamic system perturbed by Gaussian white noise (the same as our purposed model). It utilizes measurements, corrupted by Gaussian white noise, linearly related to the state and gives a statistically optimal estimator with respect to quadratic function of estimation error. The structure of the filter can be derived in a Bayesian framework as follows. At the first stage (i.e.  $t=1$ ), there’s no observation exists, thus the state vector  $x_0$  must be generated by a prior distribution that  $x_0 \sim N(\mu_0, V_0)$ , where  $\mu_0$  is the mean and  $V_0$  is the covariance matrix. By using Eq. (1), the distribution for the state vector in the first stage will be normal with parameters

$$E[x_t|y_{t-1}] = \mu_{t|t-1} = F\mu_{t-1} \tag{5}$$

$$Var[x_t|y_{t-1}] = V_{t|t-1} = FV_{t-1}F^T + \Sigma \tag{6}$$

where  $\mu_{t|t-1}$  denotes the expect value of  $x_t$  and  $V_{t|t-1}$  denotes the variance of  $x_t$  when  $y_{t-1}$  is observed. By the above information, the forecast observation would be normal distribution with parameters

$$E[y_t|y_{t-1}] = \hat{y}_t = H\mu_{t|t-1} \tag{7}$$

$$Var[y_t|y_{t-1}] = M_t = HV_{t|t-1}H^T + \gamma \tag{8}$$

The above Eqs. (5)–(8) holds for any  $t$ .

As the new observation  $y_t$  become available, the parameter vector would be updated according to Baye’s rule,

$$p(x_t|y_t) \propto p(y_t|x_t)p(x_t|y_{t-1})$$

by using Bayes' rule and standard Bayesian theory, the posterior will be normal distribution with parameters

$$\mu_{t|t} = \mu_{t|t-1} + V_{t|t-1} H^T M_t^{-1} (y_t - H \mu_{t|t-1}) \quad (9)$$

$$V_{t|t} = V_{t|t-1} - V_{t|t-1} H^T M_t^{-1} H V_{t|t-1} \quad (10)$$

After the filtering  $V_{t|t}$  and  $\mu_{t|t}$  at time  $t$ , set  $V_t = V_{t|t}$  and  $\mu_t = \mu_{t|t}$  for the next time stage. The algorithm of Kalman filter is illustrated as follows,

#### Algorithm Kalman Filter

Input:  $\mu_0, V_0, y_t := \text{observation vector}$

Output: Filtered  $\mu$  and  $V$

Begin

FOR each time step of the observation vector

Generate prediction of the new observation by

$$\hat{y}_t = H \cdot F \cdot \mu_{t-1}$$

$$M_t = H \left( F \cdot V_{t-1} \cdot F^T + \sum \right) H^T + \Gamma$$

Update the parameter by

$$\mu_{t|t} = \mu_{t|t-1} + V_{t|t-1} H^T M_t^{-1} (y_t - H \mu_{t|t-1})$$

$$V_{t|t} = V_{t|t-1} - V_{t|t-1} H^T M_t^{-1} H V_{t|t-1}$$

END FOR

END

## 2.4 Gibbs sampler

Gibbs sampler is a technique for generating random variables from a distribution indirectly, without having to calculate the density. In this paper, we make the following assumptions, (a) The initial  $x_0 \sim N(\mu_0, V_0)$ , (b) The covariance matrix  $\Sigma$  and  $\Gamma$  are known, and (c) Given  $F$ , the distribution  $x_t$  is Gaussian.

The state equation can be written

$$x_t^T = x_{t-1}^T F^T + u_t^T, \quad t = 1, 2, \dots, n \quad (11)$$

that is

$$\begin{bmatrix} x_1^T \\ \vdots \\ x_n^T \end{bmatrix} = \begin{bmatrix} x_0^T \\ \vdots \\ x_{n-1}^T \end{bmatrix} F' + \begin{bmatrix} u_1^T \\ \vdots \\ u_n^T \end{bmatrix}$$

The following notation is used for simplification.

$$X_n = \begin{bmatrix} x_1^T \\ \vdots \\ x_n^T \end{bmatrix}, X_{n-1} = \begin{bmatrix} x_0^T \\ \vdots \\ x_{n-1}^T \end{bmatrix}, U_n = \begin{bmatrix} u_1^T \\ \vdots \\ u_n^T \end{bmatrix}, F^T = \left[ F_1^T \cdots F_i^T \cdots F_p^T \right]$$

where  $F_i^T$  denotes the  $i$ th column vector of  $F^T$ . Then the Eq. (11) can be re-written as

$$X_n = X_{n-1}F^T + U_n$$

Consider the element of  $S$ , the  $p \times p$  covariance matrix being used to estimate the variance–covariance matrix of  $F$ ,

$$S(F^T) = \left\{ S_{ij} \left( F_i^T, F_j^T \right) \right\}$$

where  $S_{ij}$  denotes the  $(i, j)$  element of the matrix.  $S_{ij}$  can be calculated by the following equation.

$$\begin{aligned} S_{ij} &= \left( X_{n(i)}^T - X_{n-1}^T F_i^T \right)^T \left( X_{n(j)}^T - X_{n-1}^T \widehat{F}_j^T \right) \\ &= \left( X_{n(i)}^T - X_{n-1}^T \widehat{F}_i^T \right) \left( X_{n(j)}^T - X_{n-1}^T \widehat{F}_j^T \right) + \left( F_i^T - \widehat{F}_i^T \right) X_{n-1} X_{n-1}^T \left( F_j^T - \widehat{F}_j^T \right) \end{aligned} \tag{12}$$

where  $\widehat{F}_i^T = \left( X_{n-1} X_{n-1}^T \right)^{-1} X_{n-1} X_{n(i)}^T$  is the least square estimate of  $F_i^T$ , and  $X_{n(i)}$  is the  $i$ -th column vector of  $X_n$ . Consequently,

$$S(F^T) = A + \left( F^T - \widehat{F}^T \right)^T X_{n-1} X_{n-1}^T \left( F^T - \widehat{F}^T \right) \tag{13}$$

where  $A$  is a  $p \times p$  matrix.  $A = \{a_{ij}\}$ , where  $a_{ij}$  is the  $(i, j)$  elements of  $A$ , with

$$a_{ij} = \left( X_{n(i)}^T - X_{n-1}^T \widehat{F}_i^T \right)^T \left( X_{n(j)}^T - X_{n-1}^T \widehat{F}_j^T \right) \tag{14}$$

That means  $A$  is proportional to the sample covariance matrix. From the general result in the Gaussian model, the posterior distribution of  $F'$  is then

$$\begin{aligned} p(F^T | X) &\propto |S(F^T)|^{-\frac{n}{2}}, \quad -\infty < F^T < \infty \\ &= \left| A + \left( F^T - \widehat{F}^T \right)^T X_{n-1} X_{n-1}^T \left( F^T - \widehat{F}^T \right) \right|^{-n/2} \end{aligned} \tag{15}$$

The distribution in Eq. (15) is a matrix-variate generalization of the  $t$ -distribution. The following sampler for generating  $F^T$  and  $X$  is then proposed.

The sampling scheme generate from the conditional distributions

- a.  $x_t | F, x_{t-1}, \Sigma \sim N(Fx_{t-1}, \Sigma)$
- b.  $F^T | X, \Sigma \sim [k(n, p, p)]^{-1} |A|^{(n-p)/2} |X_{n-1} X_{n-1}^T|^{p/2} |A + FX_{n-1} X_{n-1}^T F^T|^{-n/2}$



The above sampling scheme would be the key component of the Gibbs sampler. The Gibbs sampler is a Markovian updating scheme that proceeds as follows. Given an arbitrary starting set of values  $\{Z_1^{(0)}, Z_2^{(0)}, Z_3^{(0)}, \dots, Z_k^{(0)}\}$ , and then draw

$$\begin{aligned} Z_1^{(1)} &\sim [Z_1 | Z_2^{(0)}, Z_3^{(0)}, \dots, Z_k^{(0)}], \\ Z_2^{(1)} &\sim [Z_2 | Z_1^{(1)}, Z_3^{(0)}, \dots, Z_k^{(0)}], \\ Z_3^{(1)} &\sim [Z_3 | Z_1^{(1)}, Z_2^{(1)}, \dots, Z_k^{(0)}], \dots \\ Z_k^{(1)} &\sim [Z_k | Z_1^{(1)}, Z_2^{(1)}, \dots, Z_{k-1}^{(1)}]. \end{aligned}$$

Each variable is visited in the natural order and a cycle requires  $k$  random variate generations. After  $i$  iterations we have  $(Z_1^{(i)}, Z_2^{(i)}, Z_3^{(i)}, \dots, Z_k^{(i)})$ . Under mild conditions, the following results hold (Geman and Geman 1984)

**Result 1: Convergence**

As the iteration continues,  $(Z_1^{(i)}, Z_2^{(i)}, Z_3^{(i)}, \dots, Z_k^{(i)}) \rightarrow [Z_1, Z_2, Z_3, \dots, Z_k]$ . Hence, for each sequence  $s$ ,  $Z_s^{(i)} \rightarrow [Z_s]$  as  $i \rightarrow \infty$ .

**Result 2: Rate**

Using the sup norm, the joint density of  $(Z_1^{(i)}, Z_2^{(i)}, Z_3^{(i)}, \dots, Z_k^{(i)})$  converges to the true density at a geometric rate, under visiting in the natural order.

**Result 3: Ergodic theorem**

For any measurable function  $T$  of  $Z_1, Z_2, Z_3, \dots, Z_k$  whose expectation exists,

$$\lim_{i \rightarrow \infty} \frac{1}{i} \sum_{l=1}^i T(Z_1^{(l)}, Z_2^{(l)}, Z_3^{(l)}, \dots, Z_k^{(l)}) \rightarrow E(T(Z_1, Z_2, Z_3, \dots, Z_k)).$$

For every function  $T$  on the possible configurations of the system and for every starting configuration holds with probability one.

Analytical convergence rates for Gibbs sampler applied to state space models are further discussed by Pitt and Shephard in 1996 and Robert (Pitt and Shephard 1996; Robert and Cellier 1998).

As Gibbs sampling through  $m$  replications of the aforementioned  $i$  iterations produces  $i$  independent and identically distributed  $k$  tuples  $Z_{1j}^{(i)}, Z_{2j}^{(i)}, Z_{3j}^{(i)}, \dots, Z_{kj}^{(i)}, j=1, 2, 3, \dots, m$ , which the proposed density estimate for  $[Z_s]$  having form

$$[\hat{Z}_s] = \frac{1}{m} \sum_{j=1}^m [Z_s | Z_r^{(j)}, r \neq s].$$

The above Gibbs sampling scheme on a random transition matrix and state vector forms the center part of the algorithm. In the process of generating state vectors, Kalman filtering mechanism is added. While a simple monitoring of the chain ( $Z_s$ ) can only expose strong non-stationarities, it is more relevant to consider the cumulated sums, since they need to stabilize for convergence to be achieved. The convergence control of Gibbs sampler is discussed in Section 3.

*Algorithm Gibbs Sampler*

Input:  $H:=\text{path-observation incidence matrix}$ ,  $y_t:=\text{observation sequence}$

Output:  $\hat{X}$ ,  $\hat{F}$

Begin

Initialize

$$F^{(0)} := I_p, \Sigma := I_p, \gamma := I_p$$

$$X_{store} = \{\phi\}, F_{store} = \{\phi\}$$

SET GibbsCount (g) to 0

WHILE not Converge

Generate  $x^{(g)} \sim N(\mu, V)$

Append  $x^{(g)}$  to  $X_{store}$

CALL Kalman Filter with  $\mu, V$ , and observation sequence

Generate  $F^{(g)}$  by

$$A^{(g)} = \{a_{ij}^{(g)}\}, a_{ij}^{(g)} = \left( X'_{n(i)} - X'_{n-1} \widehat{F}_i^{(g)} \right)' \left( X'_{n(j)} - X'_{n-1} \widehat{F}_j^{(g)} \right)$$

Generate  $w \sim \text{Wishart} \left( X'_{n-1} X'_{n-1} (g), n - p \right)$

Generate  $Z = \left( z'_1, z'_2, z'_3, \dots, z'_p \right), z_k \stackrel{iid}{\sim} N_p(0, A^{(g)})$

COMPUTE  $F^{(g)} = \left( \left( w^{\frac{1}{2}} \right)' \right)^{-1} Z$

APPEND  $F^{(g)}$  to  $F_{store}$

INCREMENT GibbsCount

END WHILE

READ last  $k$  items from  $X_{store}$  and put in  $X_n$

COMPUTE  $\hat{X} = \frac{1}{k} \sum X_n$

READ last  $k$  items from  $F_{store}$  and put in  $F_n$

COMPUTE  $\hat{F} = \frac{1}{k} \sum F_n$

END

2.5 Solution framework

In this subsection, the solution framework of applying Gibbs sampler and Kalman filter to the state space model is illustrated. The solution framework that combines the algorithm of Kalman filter and Gibbs sampler is demonstrated in Fig. 1. The solution framework is an iterative estimation of state vectors and transition matrix. It first filters the state vector by given transition matrix, path-observation incidence matrix (some may refer as mapping matrix), and observation vector; and then turns to estimate the transition matrix by filtered state vector. During the Kalman filter stage

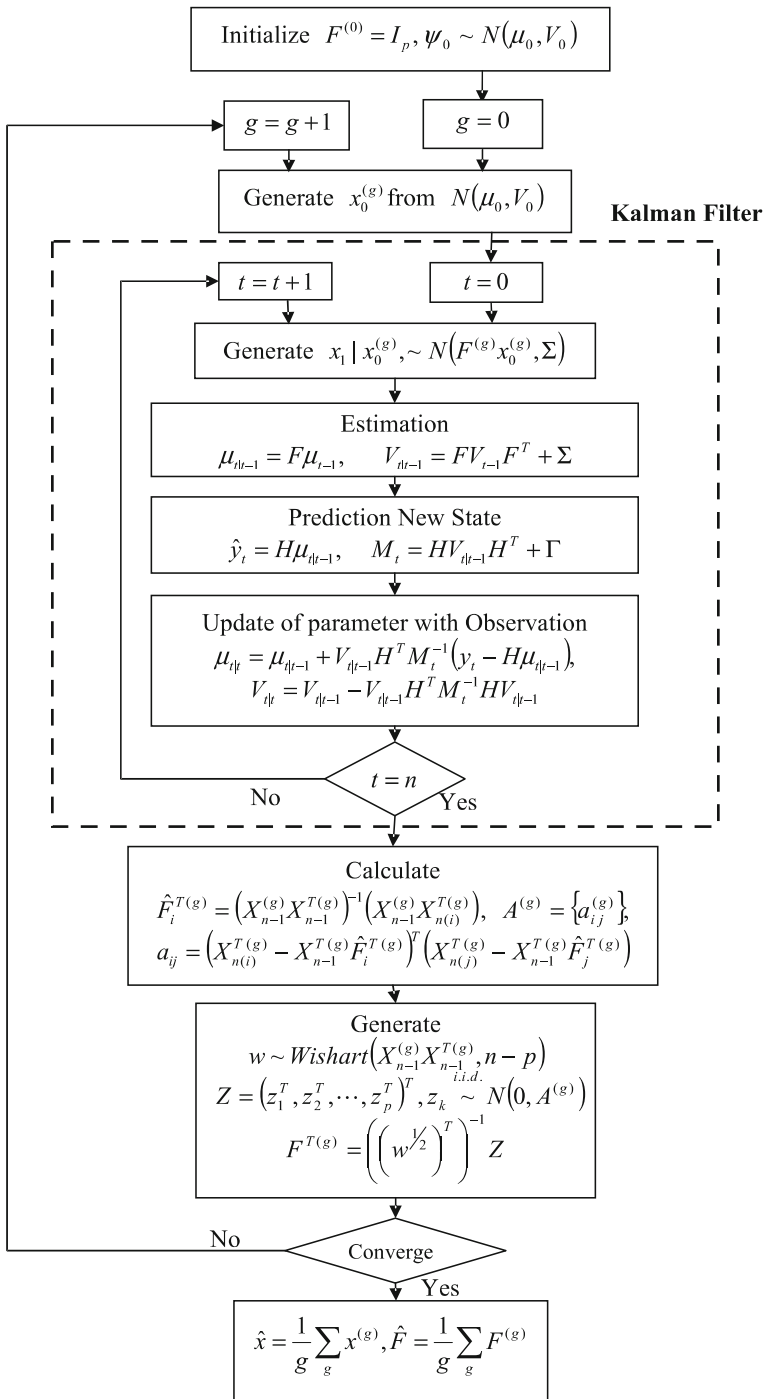


Fig. 1 The solution framework of state space model

of the framework, transition matrix and mapping matrix are fixed; while in the transition matrix estimating stage, the state vector is fixed. As the algorithm reach convergence, the transition matrix,  $\widehat{F}$ , and state vector,  $\widehat{x}$ , can be estimated.

After deriving the estimated transition matrix and state vector, prediction of state vector can be represent as,

$$\widehat{x}_{t+1} = F \cdot x_t + u_{t+1}. \tag{16}$$

And the estimation of state vector after  $h$  time periods,  $\widehat{x}_{t+h}$ , is

$$\begin{aligned} \widehat{x}_{t+h} &= F \cdot \widehat{x}_{t+h-1} + u_{t+h} \\ &= F \cdot (F \cdot \widehat{x}_{t+h-2} + u_{t+h-1}) + u_{t+h} \\ &= F \cdot (F \cdot (F \cdot (F \cdots (F \cdot \widehat{x}_t + u_{t+1}) \cdots)) + u_{t+h-2}) + u_{t+h-1} + u_{t+h} \end{aligned} \tag{17}$$

Since the expectation value of  $u$  is zero. The prediction of  $\widehat{x}_{t+h}$  can be written as

$$\widehat{x}_{t+h} = F^h x_t. \tag{18}$$

### 3 Convergence control of Gibbs sampler and parallel implementation

Gibbs sampler, an example of a Markov Chain Monte Carlo method, is an algorithm that generate sequences of samples from a joint probability distribution. It has been used to tackle a wide variety of statistical problems. If the method had been used naively without convergence control, it might result a misleading answer. Properties of applying Gibbs Sampler to state space model can be referred to Frühwirth-Schnatter (1994), Carter and Kohn (1994), Gamerman (1998), and Jungbacker and Koopman (2007); while analytic convergence rates for Gibbs sampler applied to univariate state space model can be referred to Pitt and Shephard (1996).

Theoretical guarantee of convergence is different from the practical requirement. It is thus necessary to develop a tool that determines whether the chain is converged or not. Monitoring the chain that Gibbs sampler produce is a reasonable method. However, monitoring the elements in a chain would only expose strong non-stationarities. Therefore, it is more relevant to consider the cumulated averages; since they need to stabilize for convergence to be achieved. Let the number of Gibbs iterations be denoted as  $g$ , and the corresponding element  $x^{(g)}$ . The difference between cumulated averages should be less than  $\varepsilon$ ,

$$\left| \frac{1}{g} \sum_g x^{(g)} - \frac{1}{g-1} \sum_{g-1} x^{(g)} \right| \leq \varepsilon. \tag{19}$$

Possible alternatives to empirical averages would be importance sampling, conditional expectations, or Riemann approximation techniques. However, the average may only correspond to the exploration of a single mode of the distribution by the chain; the single chain methods can never guarantee that the whole support of target distribution has been explored (Robert and Cellier 1998).

### 3.1 Multiple chain convergence of Gibbs sampler

The main idea of Gibbs sampler, an iterative simulation method, is to draw values of a random variable  $x$  from a sequence of distributions that converge to a desired target distribution of  $x$ . Single chain (sequence) methods hardly bring information on the regions of the space it does not visit. Parallel chain methods try to overcome such defect by generating parallel chains, aiming at eliminating the dependence on initial conditions. The convergence control is most often based on the comparison of the estimations of different quantities for the parallel chains. More precisely, the criterion is based on the difference between a weighted estimator of variance for each chain and the variance of the estimators on the different chains. The estimation method composed of two steps. First, create an estimate of the target distribution, centered about its mode (or modes), and over-dispersed in the sense of being more variable than the target distribution. The approximate distribution is then used to start several independent chains of the iterative simulation. The second step is to analyze the multiple chains to form a distributional estimate of the target random variable.

The monitor of convergence of the iterative simulation is to estimate the factor by which the scale of the current distribution for the target distribution might be reduced if the simulations were continued to the limit  $n \rightarrow \infty$ . This potential scale reduction can be estimated by

$$\sqrt{\hat{R}} = \sqrt{\frac{\hat{V}}{W} \frac{df}{df - 2}} = \sqrt{\left(\frac{n + 1}{n} + \frac{M + 1}{Mn} \frac{B}{W}\right) \frac{df}{df - 2}} \tag{20}$$

where

$$df = \frac{\left(\sum_{m=1}^M \frac{S_m^2}{n}\right)^2}{\sum_{m=1}^M \frac{S_m^4}{n^2(n-1)}} = \frac{\left(\sum_{m=1}^M S_m^2\right)^2 (n - 1)}{\sum_{m=1}^M S_m^4} \tag{21}$$

$\hat{R}$  declines to 1 as  $n \rightarrow \infty$ .  $\hat{R}$  is the ratio of the current variance estimate,  $\hat{V}$ , to the within-chain variance,  $W$ .  $n$  is the count of iterations,  $S_m$  are the standard deviation of each chain,  $B$  is the between-chain variance and the  $M$  is the number of parallel chains. The between- and within-chain variances,  $B$  and  $W$ , are defined as follows.

$$B = \frac{1}{M} \sum_{m=1}^M (\bar{\xi}_m - \bar{\xi})^2, \tag{22}$$

$$W = \frac{1}{M} \sum_{m=1}^M S_m^2 = \frac{1}{M} \sum_{m=1}^M \frac{1}{n} \sum_{i=1}^n (\xi_m^{(i)} - \bar{\xi}_m)^2, \tag{23}$$

with

$$\bar{\xi}_m = \frac{1}{n} \sum_{i=1}^n \xi_m^{(i)}, \quad \bar{\xi} = \frac{1}{M} \sum_{m=1}^M \bar{\xi}_m \tag{24}$$

where  $\xi_m^{(i)}$  is the  $i$ -th element in chain  $m$ .

Once  $\hat{R}$  is near 1, it is conclude that each set of the simulated values is close to the target distribution. In practice, the potential scale reduction is chosen to be 10% ( $\hat{R} \leq 1.1$ ). After it converges, we can calculate the desired sample value (the state vector) based on the empirical distribution of the  $n$  simulated iterates for each simulated chain.

### 3.2 Parallel implementation of the proposed algorithm

Gibbs sampler, a particular type of Markov Chain Monte Carlo method, requires tremendous iterations during computation; normally, there would be tens of thousands iterations in the proposed algorithm. To achieve real-time information requirement, parallel computing technique is introduced to increase the performance. The solution algorithm should be modified to adopt the parallel implementation. It is divided into several computing parts by dividing at the WHILE-LOOP. With different random seed, each computing part will lead to a different solution chain. The chain in each computing part will then be gathered to check the convergence. In this situation, communication between computing nodes is minimum, and computing power can be easily increased without communication bandwidth limitation. Figure 2 describes the parallel architecture. The parallel environment of this PC-cluster consists of 16 computing nodes; each contains two processors equivalent to Intel XEON 3.2 GHz and 1 GB memory. Nodes are connected with a Gigabits Ethernet switch for MPI protocol and a 100 Mbits PCI fast Ethernet switch for Network File System (NFS) and Network Information System (NIS).

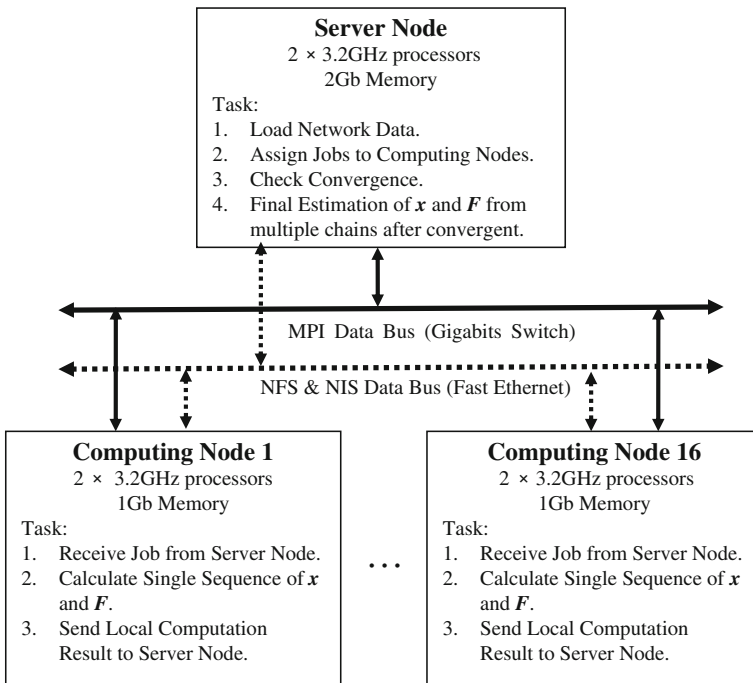


Fig. 2 The parallel environment

In the server node, parameters used in our algorithm are initialized, so does the necessary input data. When assign jobs, these input data are sent to computing nodes in the cluster through TCP/IP base intranet with Message Passing Interface (MPI) Library through gigabit switch. The computational procedure for the parallel process consists of:

1. Load input data and parameters. Initialize MPI environment.
2. Count the computing nodes exists in the cluster environment. Send data to each computing nodes.
3. Each computing nodes generate its own  $\hat{X}$  and  $\hat{F}'$  sequences by given input data. These results were sending to the server for convergence check.
4. The server check the convergence of each  $\hat{X}$  and  $\hat{F}'$ . If  $\hat{R} > 1.1$ , the computing nodes will continue step 3. As  $\hat{R} \leq 1.1$ , the server estimate the global  $\hat{X}$  and  $\hat{F}'$  by sequences generated by computing nodes.
5. Stop MPI environment. Output data.

Figure 3 shows the speedups and efficiencies of the proposed algorithm, where the speedups is the ratio of the code execution time on a single processor to that on multiple processors and efficiency is defined as the speedup divided by the number of processors. As shown in Fig. 3, the parallel scheme has efficiency of 72.5% in a 32 processors environment.

#### 4 Numerical examples

Two numerical examples are discussed in this section: (a) part of a real network of the Taipei Mass Rapid Transit (MRT) and (b) part of the Taiwan Freeway Network with Electronic Toll Collection information.

##### 4.1 Taipei Mass Rapid Transit network

A real network of Taipei MRT is tested in this section. The test network consists of nine stations in Nankang line, the topology of test network is demonstrated in

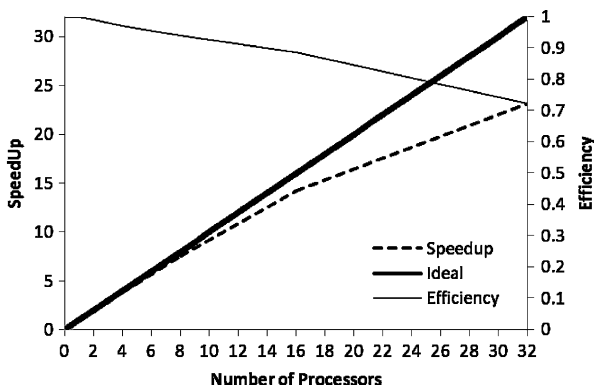


Fig. 3 Speedups and efficiencies for the parallel computing

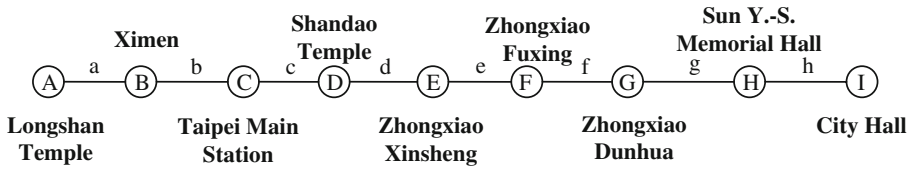


Fig. 4 The MRT test network

Fig. 4. The nodes are denoted with capital letters; while the links are denoted with lowercase letters.

Real path-flow data are the numbers of passengers traverse between stations from 18:05–20:00 in 5-minute interval; for example, there are 30 persons travel from Ximen to Taipei Main Station in the time-period of 18:00–18:35. Eight paths are included in this example defined as travelers toward Taipei Main Station from the rest of stations. These paths are demonstrated in Table 2.

In the MRT test, travel time is not considered. Therefore, link flows can be easily obtained by the summation path flows on them, i.e., flow on link *c* is equal to  $x(3)+x(4)+x(5)+x(6)+x(7)+x(8)$ , and the flow on link *e* is equal to  $x(5)+x(6)+v(7)+x(8)$ . The time-dependent link flows, illustrated in Table 3. Link flows on *b* and *c* are treated as the observation vector,  $y_t$ , in this test.

The computing environment of this test consists of four computing nodes, each nodes is responsible for a particular chain, and the server node will be responsible for,  $\hat{R}$ , the convergence check. Each particular chain has a unique random seed with difference initial values. The convergence of the estimation of the O-D pair is shown in Fig. 4 below. In convenience, only two single O-D pairs are shown; but in reality, every O-D pair must satisfy the convergence assessment. From Fig. 5(a), we might observe that after about  $2.26 \times 10^5$  iterations, four chains converge to the same point; in Fig. 5(b) it converges after about  $1.69 \times 10^5$  iterations.

After the algorithm reach convergent, transition matrix and state vector can then be estimated. The comparison of estimation output and real data is illustrated in Table 4.

Comparisons of real and estimated O-D flows by this study is measured by the value of mean absolute error (MAE)

$$MAE = \frac{\sum_{i=1}^n |x_{ESTIMATE}^i - x_{REAL}^i|}{n}. \tag{25}$$

Table 2 Path set of example MRT network

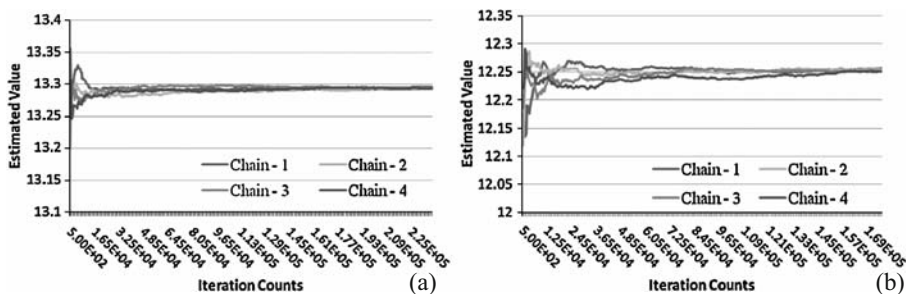
Path	O-D pair	Link set
$x(1)$	A → C	a, b
$x(2)$	B → C	b
$x(3)$	D → C	c
$x(4)$	E → C	d, c
$x(5)$	F → C	e, d, c
$x(6)$	G → C	f, e, d, c
$x(7)$	H → C	g, f, e, d, c
$x(8)$	I → C	h, g, f, e, d, c



**Table 3** Time-dependent link flows on MRT test network

Time interval	Link flow <i>a</i>	Link flow <i>b</i>	Link flow <i>c</i>	Link flow <i>d</i>	Link flow <i>e</i>	Link flow <i>f</i>	Link flow <i>g</i>	Link flow <i>h</i>
1	1	12	31	27	27	17	13	7
2	3	14	116	108	103	50	30	23
3	0	12	85	84	77	39	14	9
4	6	15	82	77	72	33	21	11
5	5	25	74	69	63	42	16	10
6	2	10	71	69	65	34	17	11
7	3	33	115	108	101	56	28	12
8	3	26	62	58	56	29	12	7
9	6	29	62	59	58	34	23	15
10	1	28	84	81	71	45	21	10
11	3	19	80	72	65	28	15	7
12	3	29	113	107	103	63	31	20
13	7	35	129	119	110	51	25	17
14	2	18	50	45	44	18	7	7
15	6	39	128	121	113	58	20	12
16	6	35	97	91	83	43	15	11
17	1	12	107	100	93	58	35	25
18	3	16	108	101	97	50	25	15
19	0	22	76	76	71	32	19	8
20	4	29	69	66	63	26	13	7
21	1	17	101	100	96	63	18	12
22	2	9	41	40	37	29	10	5
23	2	6	59	57	55	35	14	8

The value of MAE in this case is 3.97. The time-dependent differences between real and estimated O-D flows at each time interval are plotted in Fig. 6; the black line indicates the real O-D data derived from the Taipei Mass Rapid Transit company, and the grey line indicates the estimation result of the proposed algorithm. The correlation between real and estimated data is 0.898, which indicate a strong association between real and estimated data. Nearly all of the 8 O-D estimations pass the paired-samples *T* test of 95% confidence interval; only the third O-D pair has failed. We further examine the O-D pair 3 by chi-squared test, and the result leads to the conclusion of no significant difference between real and estimated data. The result shows that the proposed algorithm is capable to capture the pattern of real O-D flows.



**Fig. 5** Estimation value of the multiple chain convergence

**Table 4** The comparison of real and estimated O-D flow

Time interval	O-D pair 1		O-D pair 2		O-D pair 3		O-D pair 4		O-D pair 5		O-D pair 6		O-D pair 7		O-D pair 8	
	Real	Est.	Real	Est.	Real	Est.	Real	Est.	Real	Est.	Real	Est.	Real	Est.	Real	Est.
1	1	1.7	11	4.1	4	3.1	0	0.4	10	12.3	4	2.4	6	4.6	7	10.4
2	3	4.3	11	15	8	10.2	5	7.2	53	47.2	20	15.1	7	4.1	23	18.1
3	0	1.4	12	14.1	1	4.1	7	8.6	38	29.3	25	17.6	5	8.5	9	5.3
4	6	0.7	9	10.7	5	4	5	7.3	39	33.4	12	20.7	10	8.8	11	9.7
5	5	3.5	20	27.7	5	8.6	6	5.5	21	47.9	26	17.1	6	3.9	10	5.6
6	2	1.2	8	13.4	2	5.7	4	6.2	31	32.4	17	7.9	6	5.7	11	4.9
7	3	0.8	30	23.3	7	5	7	9.8	45	40.9	28	28.2	16	15.4	12	15.7
8	3	4.7	23	8.7	4	6	2	4.1	27	32.4	17	10.7	5	11.1	7	9
9	6	5.4	23	17.3	3	5.2	1	2.2	24	27.9	11	10.6	8	6.7	15	11.9
10	1	4.1	27	18.4	3	4.1	10	8.3	26	31.7	24	19.2	11	8.3	10	14.1
11	3	5	16	19.2	8	6.7	7	5.1	37	33.1	13	16.4	8	3.5	7	9.6
12	3	5.5	26	16.9	6	9.5	4	7.9	40	45.7	32	19.1	11	11	20	20.6
13	7	4.4	28	27.2	10	9.6	9	3.7	59	66	26	20.6	8	13	17	11.3
14	2	3.2	16	4	5	5.2	1	4.2	26	23.1	11	5.9	0	6	7	13.5
15	6	5.5	33	33.8	7	8.3	8	11.9	55	57.3	38	31.7	8	11.3	12	13.1
16	6	7.9	29	18.5	6	9.1	8	3.6	40	50.9	28	21.8	4	5.3	11	16.2
17	1	4.4	11	8.8	7	7.5	7	6.3	35	46.2	23	15.3	10	11.3	25	18.3
18	3	2.9	13	13.1	7	6.8	4	2.3	47	57.2	25	22.7	10	13.8	15	16.2
19	0	3.2	22	16.3	0	4.6	5	6.8	39	24.7	13	16.6	11	6.8	8	6.2
20	4	0.6	25	18.3	3	6.5	3	6.2	37	45	13	3.9	6	14.6	7	9.3
21	1	3.8	16	5.6	1	1.3	4	8.8	33	48.6	45	18.7	6	6.8	12	10.5
22	2	0.8	7	5.8	1	2	3	1.1	8	7.9	19	26.1	5	3.1	5	4.6
23	2	0.1	4	3.2	2	0.8	2	4.3	20	22.1	21	13.6	6	4.3	8	5.1

#### 4.2 The Taiwan freeway network with Electronic Toll Collection information

A real freeway network with Electronic Toll Collection (ETC) information is tested in this section. ETC can provide where and when the particular vehicle appeared, and helped us to trace the vehicle. The test network consists of the northern part of Taiwan freeway network, including part of freeway No.1, No. 2, and No. 3. The network is illustrated as Fig. 7. The nodes are denoted with capital letters; while the links are denoted with lowercase letters.

The test network consists of 22 O-D pairs and 30 paths, demonstrated in Table 5. Path flow data are the number of vehicles depart from their origin toll station to their destination toll station, only vehicles with ETC equipment and travel longer than two toll stations will be count. Since ETC information provides when and where a particular vehicle appears, travel time between every two stations can be calculated. Therefore, link flow on certain time can be obtained by the summation of corresponding path flows departs from their origin on particular time. For example, flow on link  $e$  at time  $t$  can be calculated by the summation of path 2, 4, 6, 8, 10 at time  $t-2$  and path 12, 14, 16, 18, 20 at time  $t-1$ . The flow count is based on ETC information on 2008/04/13.

In this subsection, transition matrix is assumed to be fixed, while path-observation incidence matrix,  $H_t$ , is generated by real data. The observation vector is time-dependent link flows, aggregated as 30-min interval, on the network. Estimated O-D

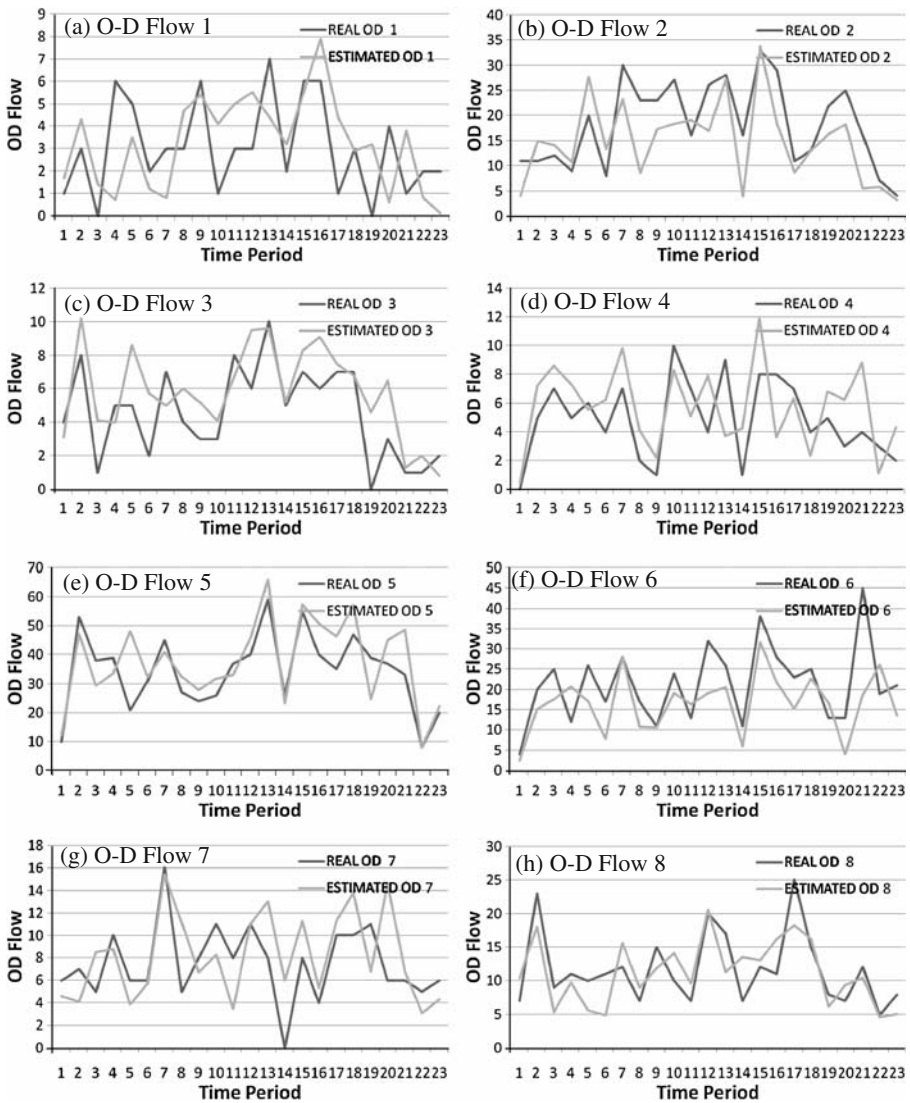


Fig. 6 The comparison of real and estimated data on MRT network

flows are obtained by multiplying path-flow with path-OD incidence matrix. The estimated O-D flows compare to O-D path flows are demonstrated in Fig. 8 as follows. In Fig. 8, the estimated, predicted, and real O-D flows are indicated in grey, dash, and black lines respectively.

The correlation between real and estimated O-D flows is 0.982, which indicates a strong association between real and estimated data. The mean absolute error is 2.41, and the mean absolute percentage error is 31.69%. Twenty out of 22 paths pass the paired-sample *T* test of 95% confidence interval; we further test the rest two paths, all of them pass the chi-square test of 95% interval.

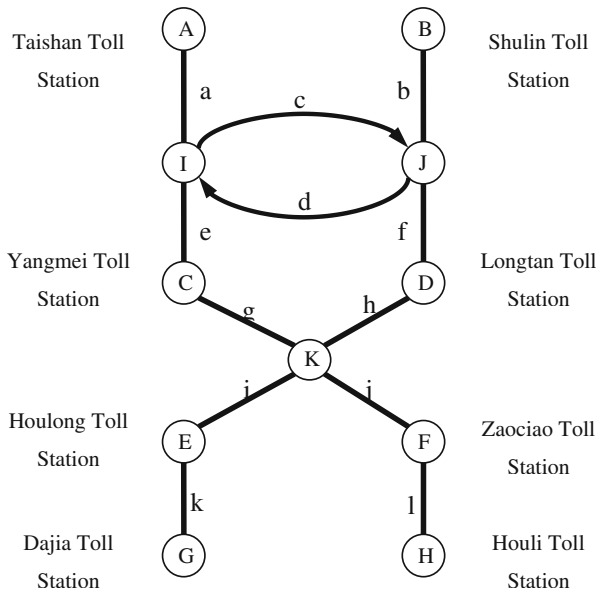


Fig. 7 The test freeway network

### 4.3 Comparison with predetermined transition matrix method

Existing researches concerning time-dependent O-D estimation usually assume the prior information of the O-D matrix (or transition matrix) is known (or at least partially known). This subsection compares the algorithm proposed in this study with predetermined transition matrix method. The example is based on the freeway network as previous section. However, the proposed algorithm no longer generates the transition matrix; it is now calculated by historical data. The transition matrix in this example is a time-dependent diagonal matrix that

$$F_t = \begin{bmatrix} \frac{\bar{x}(1)_{t+1}}{\bar{x}(1)_t} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \frac{\bar{x}(p)_{t+1}}{\bar{x}(p)_t} \end{bmatrix} \tag{26}$$

where  $\bar{x}(1)_t$  denotes the path flow of first path on time  $t$ .

The comparison of predetermined transition matrix and algorithm proposed in this research is demonstrated in Fig. 9; only O-D 1 to 4 are illustrated for simplicity. In Fig. 9, the real OD flow, flow estimated by predetermined transition matrix, and flow estimated by the proposed model are indicated in black, dash, and grey lines respectively.

The mean absolute error is 4.79, and the mean absolute percentage error is 61.87%. Predetermined transition matrix tends to preserve the O-D pattern of

**Table 5** Path set of example MRT network

Path	O-D pair	Link set
$x(1)$	B→D	b
$x(2)$	B→C	b, d, e
$x(3)$	B→E	b, f, h, i
$x(4)$	B→E	b, d, e, g, i
$x(5)$	B→G	b, f, h, i, k
$x(6)$	B→G	b, d, e, g, i, k
$x(7)$	B→F	b, f, h, j
$x(8)$	B→F	b, d, e, g, j
$x(9)$	B→H	b, f, h, j, l
$x(10)$	B→H	b, d, e, g, j, l
$x(11)$	A→D	a, c, f
$x(12)$	A→C	a, e
$x(13)$	A→E	a, c, f, h, i
$x(14)$	A→E	a, e, g, i
$x(15)$	A→G	a, c, f, h, i, k
$x(16)$	A→G	a, e, g, i, k
$x(17)$	A→F	a, c, f, h, j
$x(18)$	A→F	a, e, g, j
$x(19)$	A→H	a, c, f, h, j, l
$x(20)$	A→H	a, e, g, j, l
$x(21)$	D→E	h, i
$x(22)$	D→G	h, i, k
$x(23)$	D→F	h, j
$x(24)$	D→H	h, j, l
$x(25)$	C→E	g, i
$x(26)$	C→G	g, i, k
$x(27)$	C→F	g, j
$x(28)$	C→H	g, j, l
$x(29)$	E→G	k
$x(30)$	F→H	l

historical data; however, the O-D pattern may vary from day to day. Although predetermined transition matrix might have the advantage of preserving O-D pattern, but it might sometimes misleading the results.

## 5 Conclusions

This research presents a time-dependent origin-destination estimation method based on the state space model. Combining Gibbs Sampler and Kalman filter, we propose an algorithm loosens the assumption of known transition matrix that exists in other studies. A multiple chain convergence for the Gibbs Sampler is discussed to monitor the convergence for the proposed algorithm. To enhance the computing efficiency of this algorithm, a parallel structure is suggested and implemented on a PC-based Linux cluster. Two numerical examples are demonstrated to show the effectiveness of the proposed model. The first example is part of MRT network to address the basic model; the second example is part of freeway network to illustrate the model considering travel time. Preliminary results indicate that the proposed model is

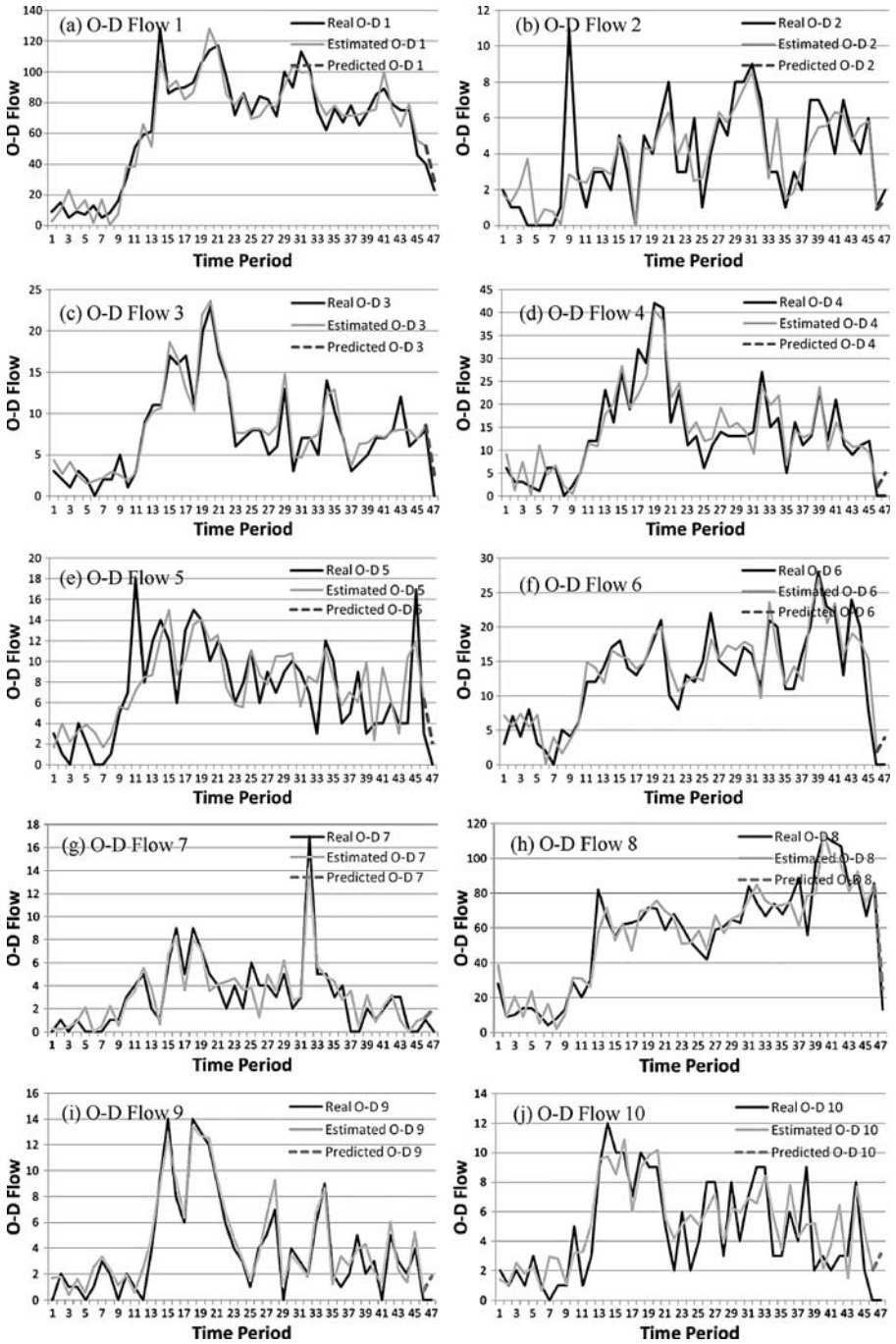


Fig. 8 The test results of freeway network

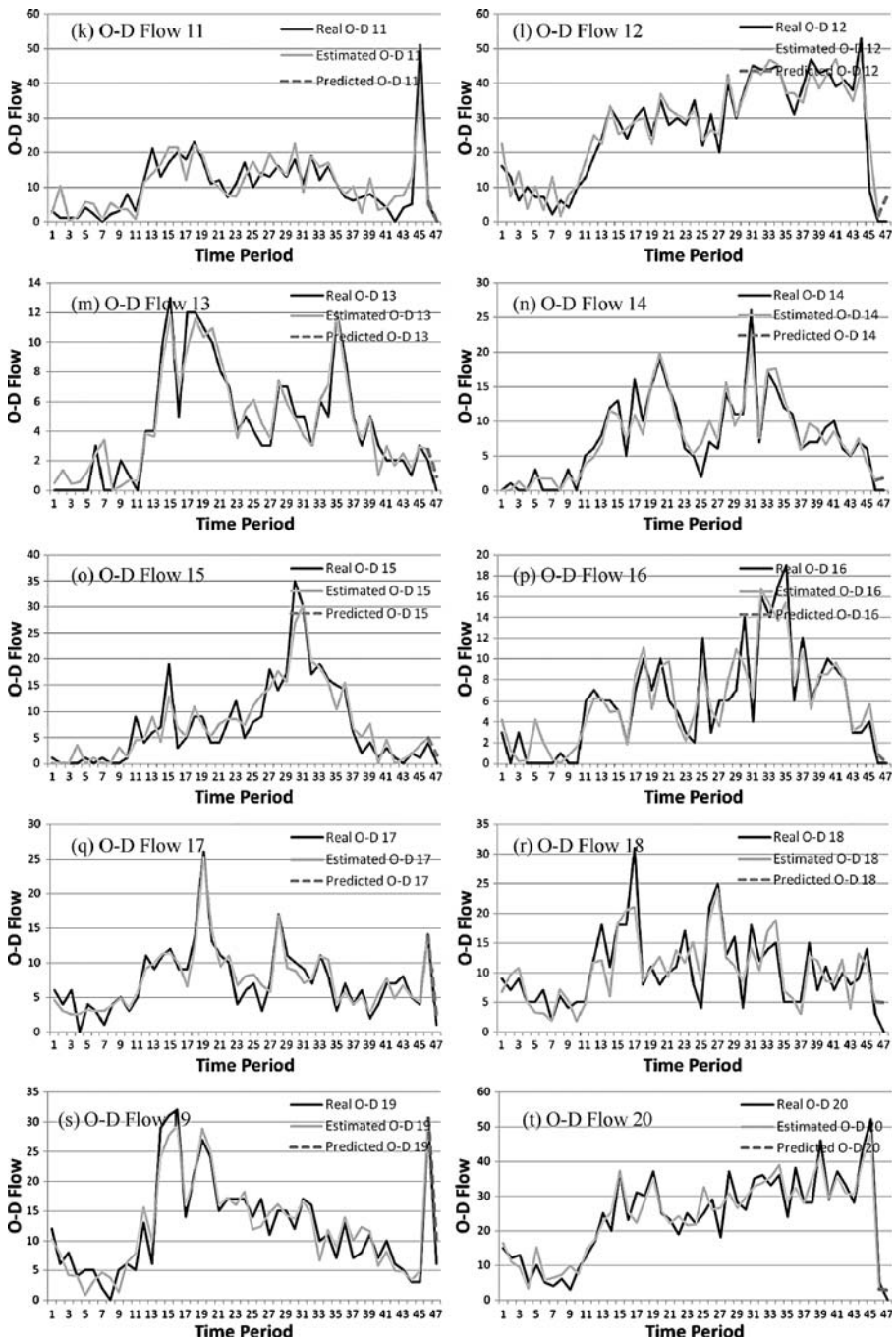


Fig. 8 (continued)

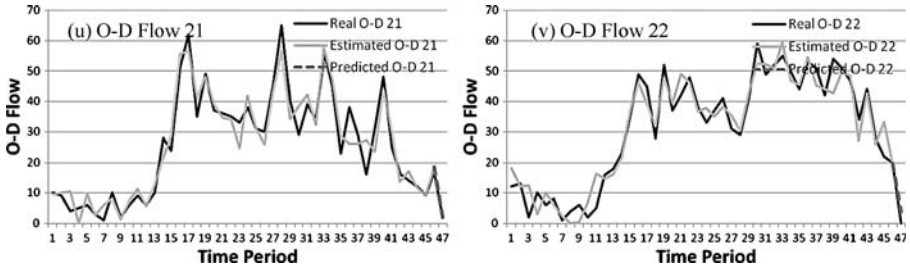


Fig. 8 (continued)

effective in the estimation of time-dependent O-D flows by inexpensive link traffic counts without prior knowledge of the O-D matrix and the transition matrix.

**Acknowledgements** This research was partially supported by the Ministry of Education of Taiwan, ROC under Grant No. EX-91-E-FA06-4-4 and the National Science Council of Taiwan, ROC under Grant No. NSC-94-2218-E-009-015. The authors would also like to thank the Far Eastern Electronic Toll Collection Co. Ltd. for ETC data.

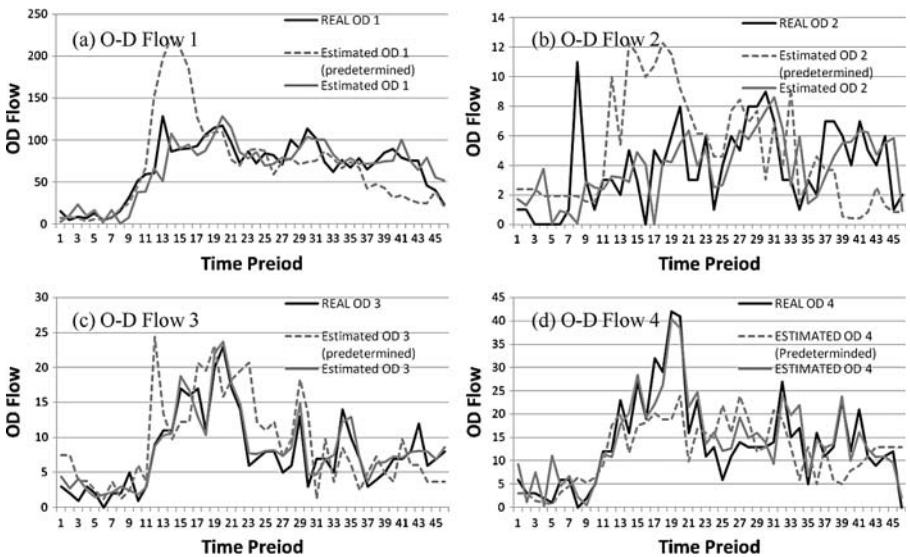


Fig. 9 The result comparison with predetermined transition matrix



## References

- Ashok K, Ben-Akiva ME (1993) Dynamic O-D matrix estimation and prediction for real-time traffic management systems. In: Daganzo CF (ed) *Transportation and traffic theory*. Elsevier, New York, pp 465–484
- Ashok K, Ben-Akiva ME (2002) Estimation and prediction of time-dependent origin-destination flows with a stochastic mapping to path flows and link flows. *Transp Sci* 36:184–198. doi:10.1287/trsc.36.2.184.563
- Bell MGH (1991) The estimation of origin-destination matrices by constrained generalized least squares. *Transp Res B* 25:13–22. doi:10.1016/0191-2615(91)90010-G
- Benamer N, Roberts JW (2004) Traffic matrix inference in IP Networks. *Netw Spatial Econ* 4:103–114. doi:10.1023/B:NETS.0000015658.75205.ed
- Bernstein D, Friesz TL, Stough R (1996) Variational inequalities, dynamical systems and control theoretic models for predicting time-varying urban network flows. *Transp Sci* 30:14–31
- Bernstein D, Friesz TL, Suo Z, Tobin RL (2001) Dynamic network user equilibrium with state-dependent time lags. *Netw Spatial Econ* 1:319–347. doi:10.1023/A:1012896228490
- Carter CK, Kohn R (1994) On Gibbs sampling for state space models. *Biometrika* 81:541–553. doi:10.1093/biomet/81.3.541
- Cascetta E (1984) Estimation of trip matrices from traffic counts and survey data: a generalized least squares estimator. *Transp Res B* 18:289–299. doi:10.1016/0191-2615(84)90012-2
- Cascetta E, Nguyen S (1988) A unified framework for estimating or updating origin/destination matrices from traffic counts. *Transp Res B* 22:437–455. doi:10.1016/0191-2615(88)90024-0
- Cascetta E, Inaudi D, Marquis G (1993) Dynamic estimators of origin-destination matrices using traffic counts. *Transp Sci* 27:363–373
- Chang GL, Tao X (1995) An advanced computing architecture for large-scale network O-D estimation. In: *Proceedings of 6th International Conference on Vehicle Navigation and Information Systems*, pp 317–327
- Chang GL, Wu J (1994) Recursive estimation of time-varying origin-destination flows from traffic counts in freeway corridors. *Transp Res B* 28:141–160. doi:10.1016/0191-2615(94)90022-1
- Cremer M, Keller H (1981) Dynamic identification of O-D flows from traffic counts at complex intersections. In: *Proceedings of 8th International Symposium on Transportation and Traffic Theory*, University of Toronto Press
- Dixon MP, Rilett LR (2005) Population origin-destination estimation using automatic vehicle identification and volume data. *J Transp Eng* 131:75–82. doi:10.1061/(ASCE)0733-947X(2005)131:2(75)
- Doblas J, Benitez FG (2005) An approach to estimating and updating origin-destination matrices based upon traffic counts preserving the prior structure of a survey matrix. *Transp Res B* 39:565–591. doi:10.1016/j.trb.2004.06.006
- Frühwirth-Schnatter S (1994) Data augmentation and dynamic linear models. *J Time Ser Anal* 15:183–202. doi:10.1111/j.1467-9892.1994.tb00184.x
- Gamerman D (1998) Markov Chain Monte Carlo for dynamic generalized linear models. *Biometrika* 85:215–227. doi:10.1093/biomet/85.1.215
- Geman S, Geman D (1984) Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images. *IEEE Trans Image Process* 6:721–741
- Jou YJ, Hwang MC (2002) Rapid transit system O/D pattern calculation with statistical Gibbs sampling and Kalman techniques. In: Taipei Rapid Transit Corp (ed) *The Proceedings of 2002 World Metro Symposium & Exhibition*
- Jou YJ, Cho HJ, Lin PW, Wang CY (2006) Incomplete information analysis for the origin-destination survey table. *J Transp Eng* 132:193–200
- Jungbacker B, Koopman SJ (2007) Monte Carlo estimation for nonlinear non-Gaussian state space models. *Biometrika* 94:827–839. doi:10.1093/biomet/asm074
- Lee JYS, Lam WHK, Wong SC (2001) Pedestrian simulation model for Hong Kong underground stations. In: *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, pp 554–558
- Lo HP, Chan CP (2003) Simultaneous estimation of an origin-destination matrix and link choice proportions using traffic counts. *Transp Res A* 37:771–788

- Maher MJ (1983) Inferences on trip matrices from observations on link volumes: a Bayesian statistical approach. *Transp Res B* 17:435–447. doi:[10.1016/0191-2615\(83\)90030-9](https://doi.org/10.1016/0191-2615(83)90030-9)
- Nie YM, Zhang HM (2008) A relaxation approach for estimating origin-destination trip tables. *Netw Spatial Econ*. doi:[10.1007/s11067-007-9059-y](https://doi.org/10.1007/s11067-007-9059-y)
- Okutani I (1987) The Kalman filter approach in some transportation and traffic problems. In: Gartner NH, Wilson NHM (eds) *Transportation and traffic theory*. Elsevier, New York, pp 397–416
- Pitt MK, Shephard N (1996) Analytic convergence rates and parameterization issues for the Gibbs sampler applied to state space models. In: *Economics Papers number 20 & 113*, Nuffield College, University of Oxford
- Robert CP, Cellier D (1998) Convergence control of MCMC algorithms. In: Robert CP (ed) *Discretization and MCMC convergence assessment*. Springer, New York, pp 27–46
- Van Zuylen HJ, Willumsen LG (1980) The most likely trip matrix estimated from traffic counts. *Transp Res B* 14:281–293. doi:[10.1016/0191-2615\(80\)90008-9](https://doi.org/10.1016/0191-2615(80)90008-9)