行政院國家科學委員會專題研究計畫　成果報告

子計畫一：動態旅次起訖推估與預測(II)

計畫類別：整合型計畫
計畫編號：NSC94-2218-E-009-015-
執行期間：94 年 08 月 01 日至 95 年 07 月 31 日
執行單位：國立交通大學統計學研究所

計畫主持人：周幼珍

報告類型：精簡報告
報告附件：出席國際會議研究心得報告及發表論文
處理方式：本計畫可公開查詢

中　華　民　國 95 年 11 月 5 日

Dynamic Origin-Destination Estimation and Its Parallel Implementation

Yow-Jen Jou, David Bernstein, Chien-Lun Lan

***Abstract* —**Origin-Destination (O-D) information is important in many transportation related domains, such as transportation planning, urban and regional planning, traffic assignment, and so on. Historical studies assumed that the transition matrix is known or at least approximately known, which is unrealistic for a real world network. To obtain real-time O-D information in a reasonable way, state space model with Gibbs sampler and Kalman filter is then introduced by researchers. The Gibbs sampler method mentioned above requires considerable quantities of iterations and takes massive computation time, thus parallel computing method has been introduced to increase the computing power. This paper implements parallel computation for the O-D estimation algorithm on a Linux cluster and gives a numerical example which shows a satisfying result.

***Keywords* —** dynamic origin-destination, state space model, Gibbs sampler, traffic, PC cluster

1. Introduction

Origin-Destination (O-D) information is in many transportation related domains, such as transportation planning, urban and regional planning, traffic assignment, and even pedestrian simulation (Bernstein, 1996, 1997, 2001; Chang, 1994, 1999, 2004, 2005; Jou, 2001,2002; Gao 2002; Lee 2001). The O-D matrix provides a planner with the basic information for planning a transportation system (Nanda, 1993). Traditional ways of acquire actual O-D information can be obtained by an extensive traffic survey, including license plate recognizing, automatic vehicle identification, and so on; this is not only a cumbersome process but also very costly. Real-time O-D information, which plays an important role in Intelligent Transportation System (ITS), is not possible to obtain by the traditional method. With real-time O-D information, numerous high-value ITS applications e.g., just-in-time delivery, time shortest path emergency vehicle routing and congestion avoidance would be feasible. Due to this reason, researchers have been seeking estimation methods to derive valuable O-D flow information from less expensive traffic data, mainly link traffic counts of surveillance systems. The existing studies on this subject can be roughly divided into two categories: 1) Assignment-Based Methods and 2) Non-Assignment-Based Methods (Chang, 1995).

The concept of assignment-based methods is mainly extended from static O-D estimation method. These methods based on the assumption of reliable descriptive model for network flow assignment. The dynamic models developed by this approach appears to address the complex time-dependent issues in urban networks effectively, if the descriptive model mentioned above exists and a time-series set of actual O-D information is available. Another approach is the non-assignment based method. The key concept of this approach is to assume that each O-D parameter remains approximately constant during consecutive intervals of measurements. Researchers proposed a variety of estimation methods in this subject, and the primary difference among all recursive algorithms for dynamic O-D estimation lies in their unique way of computing filter matrix. The main strength of non-assignment-based method is the capability of providing O-D information from surveillance system which would be relatively accurate and less expensive cost.

Jou introduced state pace model into dynamic O-D estimation which estimates both O-D matrices and transition matrix simultaneously without any prior information of state variables, 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. Gibbs sampler, a particular type of Markov Chain Monte Carlo (MCMC) method, has been introduced in the solution algorithm to overcome the shortcoming of the assumption of known transition matrix. The solution algorithm requires considerable quantities of iterations and takes massive computation time, thus parallel computing technique is then been introduced to improve the performance.

The remainder of this paper is organized as follows, the dynamic origin-destination estimation (DODE) by state space model is introduced in section 2, the parallel implementation of DODE is addressed in section 3, and the conclusion is outlined in the last section.

2. Dynamic Origin-Destination Estimation

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

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

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, Σ is the corresponding covariance matrix. x the state variable, is defined to be the path flow belonging to an O-D pair.

Next, the observation equation,

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

where y_t is the $q \times 1$ observation vector which means there are q detectors on the road network. The number of O-D pairs is denoted by *p*. *H* is a $q \times p$ zero-one matrix, which denotes routing matrix for a network. v_t is also a noise term that $v_t \sim N_q(0, \Gamma)$. 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_t by latest available information.

There are two major elements to be incorporated in the solution method, 1) filtering states by observations, and 2) sampling scheme of *F* and state variables. Since the observations y_t are not used in the conditional distribution, the Kalman filter and the Gibbs sampler must be combined.

2.1 Kalman Filter

The Kalman filter is a system of recursions to estimate the state vector before a new observation arrives, to forecast the observation, and to update the state vector as the new observation is known. The structure of the filter can be derived in a Bayesian framework as follows. 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 $x_0 \sim Normal(\mu_0, V_0)$, where μ_0 is the mean and V_0 is the covariance matrix. By using equation (1), the distribution for the state vector in the first stage will be normal with parameters

$$E[x_t \mid y_{t-1}] = \mu_{t|t-1} = F\mu_{t-1} \quad (3)$$

$$Var\left[x_t \mid y_{t-1}\right] = V_{t|t-1} = FV_{t|t-1}F' + \Sigma \quad (4)$$

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

$$E\left[y_t \mid y_{t-1}\right] = \hat{y}_t = H\mu_{t|t-1} \quad (5)$$

$$Var\left[y_t \mid y_{t-1}\right] = M_t = HV_{t-1}H' + \Upsilon \quad (6)$$

The above equation (3)-(6) holds for any *t*.

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

$$p\left(x_t \mid y_t\right) \propto p\left(y_t \mid x_t\right) p\left(x_t \mid y_{t-1}\right)$$

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

$$\mu_{t|t} = \mu_{t|t-1} + V_{t|t-1}H'M_t^{-1}\left(y_t - H\mu_{t|t-1}\right) \quad (7)$$

$$V_{t|t} = V_{t|t-1} - V_{t|t-1}H'M_t^{-1}HV_{t|t-1} \quad (8)$$

The algorithm of Kalman Filter is illustrated as follows,

Algorithm Kalman Filter

Input: μ_0 , V_0 , $y_t := observation\ sequence$
Output: Filtered μ and *V*
Begin
FOR each time step of the observation sequence
Generate prediction of the new observation by
$\hat{y}_t = H \cdot F \cdot \mu_{t-1}$

$$M_t = H \cdot \left(F \cdot V_{t-1} \cdot F' + \Sigma\right) \cdot H' + \Upsilon$$

Update the parameter by

$$\mu_{t|t} = \mu_{t|t-1} + V_{t|t-1} H' M_t^{-1} \left(y_t - H \mu_{t|t-1}\right)$$

$$V_{t|t} = V_{t|t-1} - V_{t|t-1} H' M_t^{-1} H V_{t|t-1}$$

END FOR

END

2.2 Gibbs Sampler

The 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, 1) The initial $x_0 \sim N\left(\mu_0, V_0\right)$, 2) The covariance matrix Σ and Υ are known, and 3) Given *F*, the distribution x_t is Gaussian.

The state equation can be written

$x'_t = x'_{t-1} F' + \mu'_t$, *t*=1,2,…,n

that is

$$\begin{bmatrix} x'_1 \\ \vdots \\ x'_n \end{bmatrix} = \begin{bmatrix} x'_0 \\ \vdots \\ x'_{n-1} \end{bmatrix} F' + \begin{bmatrix} \mu'_1 \\ \vdots \\ \mu'_n \end{bmatrix}$$

The following notation is used for simplification.

$$X_n = \begin{bmatrix} x'_1 \\ \vdots \\ x'_n \end{bmatrix}, \quad X_{n-1} = \begin{bmatrix} x'_0 \\ \vdots \\ x'_{n-1} \end{bmatrix}, \quad U = \begin{bmatrix} \mu'_1 \\ \vdots \\ \mu'_n \end{bmatrix}, \quad F' = \begin{bmatrix} F' & \cdots & F' & \cdots & F' \end{bmatrix}$$

Consider the element of the symmetric matrix

$$S\left(F'\right) = \left\{S_{ij}\left(F'_i, F'_j\right)\right\}$$

$$
\begin{aligned}
S_{ij}\left(F_i', F_j'\right) &= \left(X'_{n(i)} - X'_{n-1}F_i'\right)' \left(X'_{n(j)} - X'_{n-1}F_j'\right) \\
&= \left(X'_{n(i)} - X'_{n-1}\hat{F}_i'\right)' \left(X'_{n(j)} - X'_{n-1}F_j'\right) + \left(F_i' - \hat{F}_i'\right)' X_{n-1}X'_{n-1}\left(F_j' - F_j'\right)
\end{aligned}
$$

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

$$S\left(F'\right) = A + \left(F_i' - \hat{F}_i'\right)' X_{n-1}X'_{n-1}\left(F_j' - F_j'\right)$$

where A is a $p \times p$ matrix, $A = \left\{a_{ij}\right\}$ with

$$a_{ij} = \left(X'_{n(i)} - X'_{n-1}\hat{F}_i'\right)' \left(X'_{n(j)} - X'_{n-1}F_j'\right)$$

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

$$
\begin{aligned}
p\left(F' \mid X\right) &\propto \left|S\left(F'\right)\right|^{-n/2}, \quad -\infty < F' < \infty \\
&= \left|A + \left(F_i' - \hat{F}_i'\right)' X_{n-1}X'_{n-1}\left(F_j' - F_j'\right)\right|^{-n/2}
\end{aligned}
\qquad (9)
$$

The distribution in equation (9) is a matrix-variate generalization of the t-distribution. The following sampler for generating F' and X is then proposed.

Sampling Scheme *Generate from the conditional distributions*

a. $x_t \mid F, x_{t-1}, \Sigma \sim N\left(Fx_{t-1}, \Sigma\right)$

b. $F' \mid X, \Sigma \sim \left[k\left(n, p, p\right)\right]^{-1} \left|A\right|^{(n-p)/2} \left|X_{n-1}X'_{n-1}\right|^{p/2} \left|A + FX_{n-1}X'_{n-1}F'\right|^{-n/2}$

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 $Z_1^{(0)}, Z_2^{(0)}, Z_3^{(0)}, \ldots, Z_k^{(0)}$, and then draw $Z_1^{(1)} \sim \left\lfloor Z_1 \middle| Z_2^{(0)}, Z_3^{(0)}, \ldots Z_k^{(0)} \right\rfloor$, $Z_2 \sim \left\lfloor Z_2 \middle| Z_1^{(0)}, Z_3^{(0)}, \ldots Z_k^{(0)} \right\rfloor$, and so on. Each variable is visited in the natural order and a cycle requires *k* random variate generations. After *i* iterations we have $(Z_1^{(i)}, Z_2^{(i)}, Z_3^{(i)}, ..., Z_k^{(i)})$. Under mild conditions, Geman and Geman showed that the following results holds(Robert, 1998).

Result 1 Convergence

$(Z_1^{(i)}, Z_2^{(i)}, Z_3^{(i)}, ..., Z_k^{(i)}) \to [Z_1, Z_2, Z_3, ..., Z_k]$ and hence for each *s*, $Z_s^{(i)} \to [Z_s]$ as $i \to \infty$. In fact a slightly stronger result is proven. Rather than requiring that each variable be visited in repetitions of the natural order, convergence still follows any visiting scheme, provided that each variable is visited infinitely often.

Result 2 Rate

Using the sup norm, rather than the L_1 norm, the joint density of $(Z_1^{(i)}, Z_2^{(i)}, Z_3^{(i)}, ..., Z_k^{(i)})$ converges to the true density at a geometric rate in *i*, under visiting in the natural order.

Result 3 Ergodic theorem

For any measurable function *T* of $Z_1, Z_2, Z_3, ..., Z_k$ whose expectation exits,

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

As Gibbs sampling through *m* replications of the aforementioned *i* iterations produces *k* tuples $\left(Z_{1j}^{(i)}, Z_{2j}^{(i)}, Z_{3j}^{(i)}, ..., Z_{kj}^{(i)}\right)\left(j = 1,2,3,...,m\right)$, which the proposed density estimate for $[Z_s]$ having form $\left[\hat{Z}_s\right] = \frac{1}{m} \sum_{j=1}^{m} \left[Z_s \middle| Z_r^{(j)}, r \neq s\right]$.

The above Gibbs sampling scheme on a random transition matrix and state variable forms the center part of the algorithm. In the process of generate state variables, Kalman filtering mechanism is added. While a simple monitoring of the chain ($x^{(g)}$) 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 (Robert 1998). The solution algorithm with the time-complexity of $O\left(n^2\right)$ is shown as follows,

Algorithm Gibbs Sampler

Input: $H := path - observation\ incidence\ matrix$, $y_t := observation\ sequence$
Output: $\hat{X}$, $\hat{F}$

Begin

Initialize

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

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

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 μ, V, and observation sequence

Generate $F'^{(g)}$ by

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

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

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

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

Accurate O-D information is difficult to obtain on a real road network, thus other O-D information must be considered for validation. A numerical test of the DODE algorithm with the real data has been conducted by Jou (Jou, 2003). The algorithm is implemented on the Mass Rapid Transit network in Taipei, which consists of nine stations. The results are generally satisfactory, showing that also in the unknown transition matrix case, significant estimates could be obtained.

3. The Implementation of Parallel Computation and its Results

Computation power is crucial to achieve real-time information requirement, thus the parallel computing technique is then been introduced to satisfy this requirement. The solution algorithm should be modified to adopt the parallel implementation. It is then divided into several computing parts by dividing it 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 1 describes the parallel architecture; a similar architecture had been proposed by Li (Li, 2002). In the pre-processor section, 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. The computational procedure for the parallel process consists of:

Step 1. Load input data and parameters. Initialize MPI environment.

Step 2. Count the computing nodes exits in the cluster environment. Decide the count of samples should be generated by each computing nodes. Send data to each computing nodes.

Step 3. Each computing nodes generate its own $\hat{X}$ and $\hat{F}'$ by given input data for given times. And then send the result to server.

Step 4. After all the data been sent to server, the server check the convergence by each $\hat{X}$ and $\hat{F}'$. If converge, the server estimate the global $\hat{X}$ and $\hat{F}'$ by averaging $\hat{X}$ and $\hat{F}'$ of each computing node.

Step 5. Stop MPI environment. Output data.

An empirical test is conducted on a part of real network located in Taiwan, the HsinChu Science-based Industrial Park. The network is a closed network, with 11 entrances, which consists of 91 nodes and 244 links, shown as figure 2. There exist 17 observation sites on the network; the observation site updates traffic count every minute.

The parallel environment of this research consists of 16 computing nodes; each contains 2 Intel XEON 3.2GHz processors and 1 GB memory. Nodes are connected with a 1 Gigabits Ethernet switch for MPI protocol and a 100 Mbits PCI fast Ethernet switch for Network File System (NFS) and Network Information System (NIS). Figure 3 shows the speedups and efficiencies, 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(Gropp, 1999; El-Rewini, 1998), of the parallel computing for 100 samples on the 32 CPU Linux-cluster with MPI library. As shown in Figure 3, a quite good value of the speedup and efficiency of the parallel scheme is achieved. That means we can decrease the computation time easily to achieve the goal of real-time information.

4. Conclusions

This paper introduced a dynamic origin-destination estimation method by state space model. With Gibbs Sampler and Kalman filter, the algorithm loosens the assumption of known transition matrix that exists in other studies. To satisfy the real-time computation requirement of this algorithm, a parallel implementation on a PC-based Linux cluster is conducted. The parallel implementation shows a good result of nearly 80% of computing power remains for each CPU under a 32 CPUs cluster environment. That leads to the conclusion: real-time estimation of O-D matrices can be achieved by increasing a reasonable amount of CPUs.

Acknowledgement

This research was supported in part by the Ministry of Education of Taiwan, ROC under Grants EX-91-E-FA06-4-4 and in part by the National Science Council of Taiwan, ROC under Grants NSC-94-2218-E-009-015 respectively.

Authors

Yow-Jen Jou, Institute of Statistics, National Chiao-Tung University, Taiwan, yjjou@stat.nctu.edu.tw

David Bernstein, College of Integrated Science and Technology, James Madison University, bernstdh@jmu.edu

Chien-Lun Lan, Department of Transportation Technology and Management, National Chiao-Tung University, Taiwan, blue.tem92g@nctu.edu.tw

References

[1] C. K. Cater and R. Kohn, "*Markov Chain Monte Carlo in Conditionally Gaussian State Space Models*", Biometrika, Vol. 83, No. 3, pp.589-601, 1996.

[2] Christian P. Robert, "*Discretization and MCMC Convergence Assessment*", Springer, ISBN: 0-387-98591-3, 1998.

[3] D.A. bader, J. JaJa, R. Chellappa, "*Scalable data parallel algorithms for texture synthesis using Gibbs random fields*", IEEE Transactions on Image Processing, Vol. 4, Issue 10, 1995.

[4] Daniel Sorensen, "*Likelihood, Bayesian and MCMC methods in quantitative genetics*", Springer-Verlag, 2002.

[5] David Bernstein, T.L. Friesz, Z. Suo and R.L. Tobin, "*Dynamic Network User Equilibrium with State-Dependent Time Lags*", Networks and Spatial Economics, Vol. 1, pp. 319-347, 2001.

[6] David Bernstein, T.L. Friesz, "*Infinite Dimensional Formulations of Some Dynamic Traffic Assignment Models*", Network Infrastructure and the Urban Environment, Springer, New York, pp.112-124, 1997.

[7] David Bernstein, T.L. Friesz and R. Stough, “*Variational Inequalities, Dynamical Systems and Control Theoretic Models for Predicting Time-Varying Urban Network Flows*”, Transportation Science, Vol.30, pp.14-31, 1996.

[8] Gang-Len Chang, Xianding Tao, “*Advanced computing architecture for large-scale network O-D estimation*”, Proceedings of the 6th 1995 Vehicle Navigation and Information System Conference, pp.317-327, 1995.

[9] Gnag-Len Chang, J. Wu, “*Recursive estimation of time-varying origin-destination flows from traffic counts in freeway corridors*”, Transportation Research 28B, pp.141-160, 1994.

[10] Gnag-Len Chang, X.D. Tao, “*An integrated model for estimating time-varying network origin-destination distributions*”, Transportation Research, Vol. 33A pp.381-399, 1999.

[11] H. El-Rewini, T. G. Lewis, “*Distributed & Paralle Computing.*”, Manning, Greenwich, NY, 1998.

[12] Iljoon Chang, Gang-Len Chang, ”*Calibration of Intercity Trip Time Value with a Network Assignment Model: A Case Study for the Korean NW-SE Corridor*”, Journal of Transportation and Statistics, Vol. 5, 2002.

[13] Jodie Y. S. Lee, William H. K. Lam, and S. C. Wong, “*Pedestrian Simulation Model for Hong Kong Underground Stations*”, Proceeding of the IEEE

International Conference on Intelligent Transportation Systems, pp. 554-558, 2001.

[14] John H. Mathews, "*Numerical Methods*", Prentice-Hall, ISBN: 0-13-626656-8, 1987.

[15] Pei-Yih Ting, R.A. Iltis, "*Diffusion network architectures for implementation of Gibbs samplers with applications to assignment problems*", IEEE Transactions on Neural Networks, Vol. 5, Issue: 4, 1994.

[16] Pei-Wei Lin, Gang-Len Chang, Saed Rahwanji, "*Robust Model for Estimating Freeway Dynamic Origin-Destination Matrix*", Transportation Research Board, 2005.

[17] Pierre Bremaud, "*Markov chains: Gibbs fields, Monte Carlo simulation, and queues*", Springer, 1999.

[18] Raman Nanda, and Shinya Kikuchi, "*Estimation of Trip O-D Matrix When Input and Output are Fuzzy*", Proceedings of the IEEE International Symposium on Uncertainty Modelling and Analysis, pp. 104-111, 2003.

[19] Song Gao, and Ismail Chabini, "*Policy-Based Stochastic Dynamic Traffic Assignment Models and Algorithms*", Proceeding of the IEEE International Conference on Intelligent Transportation Systems, pp. 445-453, 2002.

[20] W. Gropp, E. Lusk, R. Thakur, "*Using MPI-2: Advanced Features of the*

Message-Passing Interface.", MIT Press, Cambridge, MA, 1999.

[21] Yow-Jen Jou, Ming-Chorng Hwang and Jia-Ming Yang, "*A Traffic Simulation Interacted Approach for the Estimation of Dynamic Origin-Destination Matrix*", IEEE International Conference on Networking, Sensing and Control, Vol. 2, 2004.

[22] Y. Li, S.M. Sze and T.-S. Chao, "*A Practical Implementation of Parallel Dynamic Load Balancing for Adaptive Computing in VLSI Device Simulation*", Engineering with Computers, Vol. 18, pp. 124-137, 2002.

[23] Y.J. Jou, M.C. Hwang, Y.H. Wang, and C.H. Chang, "*Estimation of Dynamic Origin-Destination by Gaussian State Space Model with Unknown Transition Matrix*," 2003 IEEE International Conference on Systems, Man & Cybernetics, 5-8 Oct. 2003, Washington, D.C., USA.

[24] Yow-Jen Jou, Ming-Chorng Hwang, "*Rapid Transit System Origin-Destination Matrix Estimation*", 2002 World Metro Symposium & Exhibition, Taipei, April 2002.

[25] Yow-Jen Jou, "*Rapid Transit System O/D Pattern Calculation with Statistical Gibbs Sampling and Kalman Techniques*", Conference on Computational Physics 2001, 5-8 September, 2001, Germany.

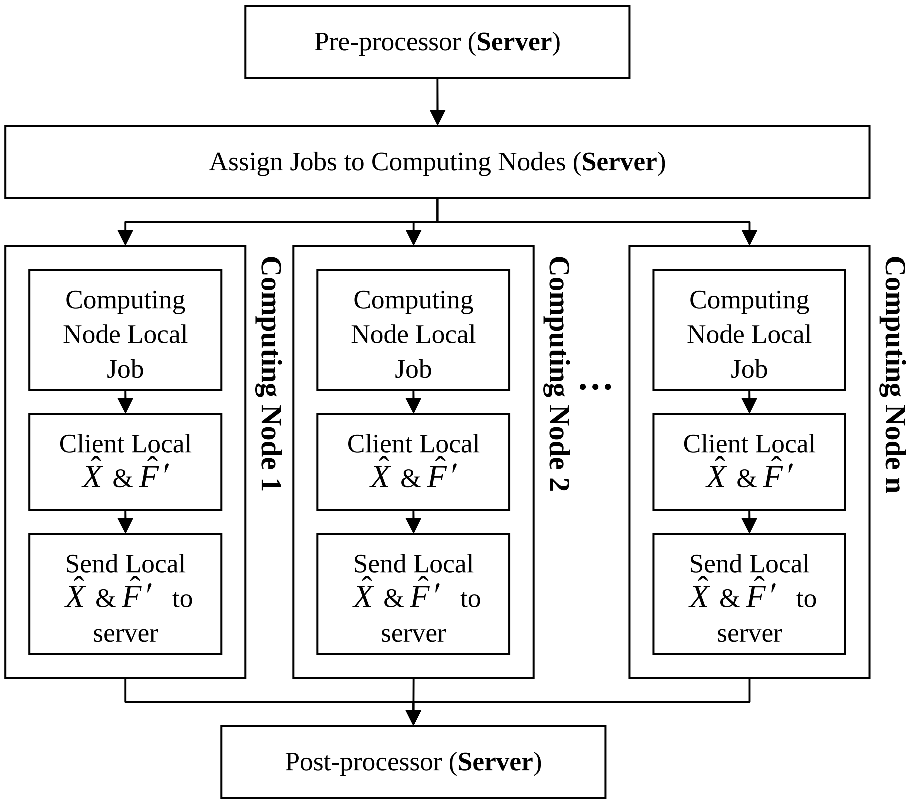

Figure 1. The flow chart of parallel algorithm

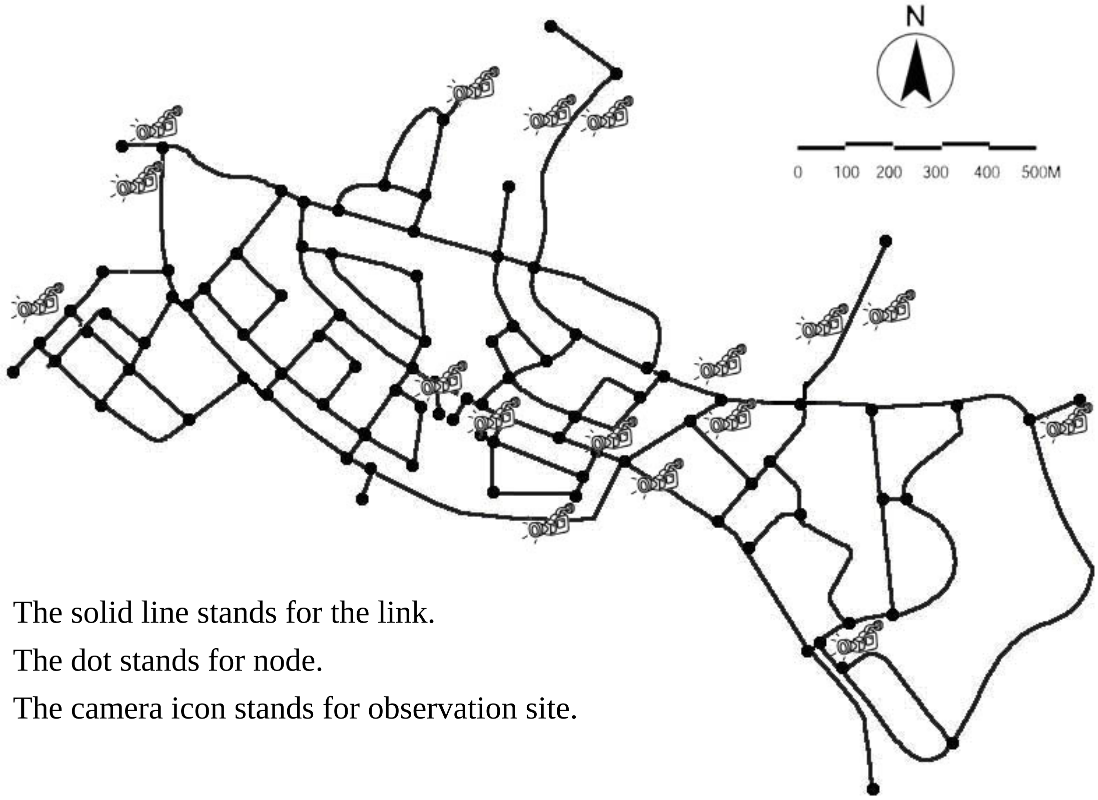

Figure 2. The test network.

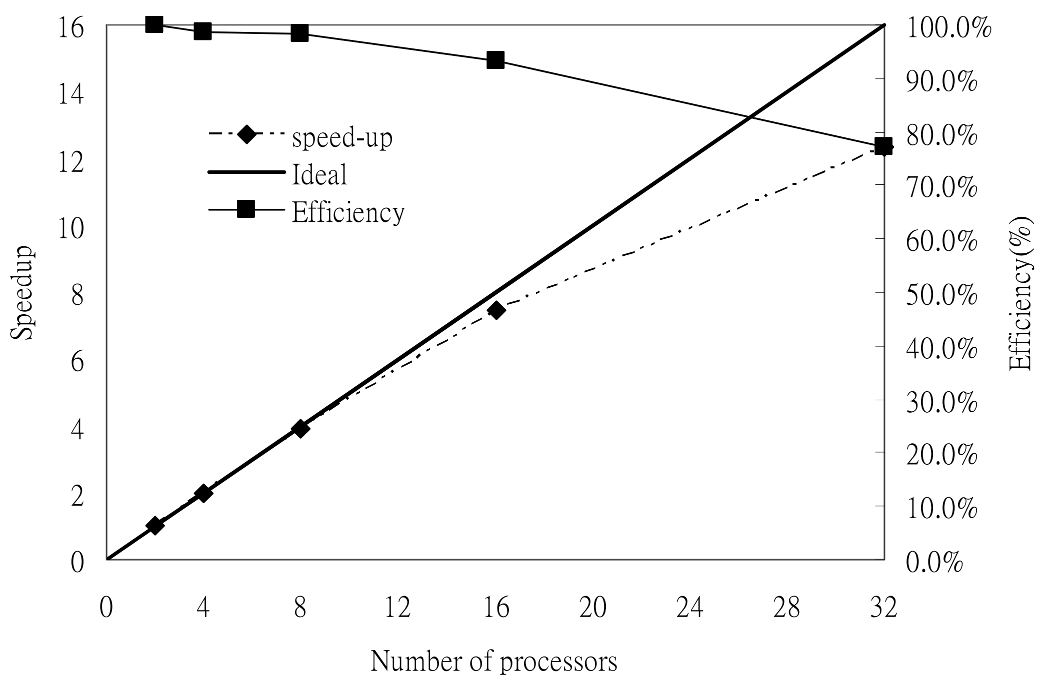

Figure 3. Speedups and efficiencies for the parallel computing