行政院國家科學委員會專題研究計畫　成果報告

於交通量指派問題之重力式互動性馬可夫模式中放鬆獨立且無關選項限制之研究

計畫類別：個別型計畫
計畫編號：NSC93-2415-H-009-012-
執行期間：93 年 08 月 01 日至 94 年 07 月 31 日
執行單位：國立交通大學運輸科技與管理學系(所)

計畫主持人：謝尚行

報告類型：精簡報告

處理方式：本計畫可公開查詢

中　華　民　國 94 年 9 月 8 日

於交通量指派之之重力式互動性馬可夫模式中

放鬆獨立且無關選項限制之研究

Shang-Hsing Hsieh[1], Jan-Mou Li[2]

中文摘要

重力式互動性馬可夫鏈(Gravity-Type Interactive Markov model：GIM model)為 Smith 與 Hsieh(1994, 1997)所提出，將此模式用於隨機交通量指派問題中，可建立隨機交通量指派問題之 GIM 模式(簡稱 GIMT 模式)。在方案為獨立且非相關(IIA)的假設前提下，謝尚行與董珈汶(民國 87 年)證明了 GIMT 模式的穩定狀態條件等同於路網之隨機使用者均衡(SUE)條件。但在真實路網中，各備選路徑間多半都包含一些重覆路段(overlapping links)，這使得 GIMT 模式的應用受到很大的限制。

文獻中探討放鬆羅吉特模式中 IIA 條件的研究非常豐富，本研究將參考 McFadden (1978)、Ben-Akiva and Lerman(1985)、Börsch-Supan(1990)等以巢式羅吉特(Nested Logit)及 Chu(1981)、Koppelman, F.S. and Chieh-Hua Wen(1998, 2000)等以配對組合羅吉特(Paired Combinatorial Logit, PCL)放鬆 IIA 條件的理論，探討於 GIM 模式中放鬆 IIA 條件的可行性及具體做法，從而建立放鬆 IIA 條件的 GIMT 演算法，以求解路徑具重覆路段之路網的 SUE，並以範例說明之。

關鍵詞：重力式互動性馬可夫鏈、隨機使用者均衡、巢式羅吉特、配對組合羅吉特

英文摘要

Gravity-Type Interactive Markov model was introduced by Smith and Hsieh (1994, 1997). A model so called GIMT can be formulated as applying the GIM model to stochastic traffic assignment problems. Under the assumption of independent and irrelevant alternatives (IIA), Hsieh and Dong (1998) proved that the steady-state condition of a GIMT model is equivalent to the condition of stochastic user equilibrium (SUE) in a network. However, most paths in actual networks have some links overlapped. This makes great limitation to the application of GIMT model.

Previous researches devote great efforts in relaxing the IIA assumption of logit choice model. One of them is nested logit which was introduced by McFadden(1978), Ben-Akiva and Lerman(1985) and Börsch-Supan(1990). Another is paired combinatorial logit (PCL) which was introduced by Chu(1981) and Koppelman, F.S. and Chieh-Hua Wen(1998, 2000). We attempt to employ these theories in relaxing the

IIA assumption of GIM model. Then we will try to establish an IIA-relaxed GIMT algorithm to solve the stochastic user equilibrium in networks having some paths overlapped. Illustrative examples will also be presented.

Keywords: Gravity-Type Interactive Markov Model, Stochastic User Equilibrium (SUE), Nested Logit, Paired Combinatorial Logit

1. Introduction

Smith and Hsieh first introduced the gravity-type interactive Markov model (1994, 1997). It is an interactive Markov model. The class of interactive Markov models first introduced by Matras (1967) and Conlisk (1976) has been widely studied and applied in the social sciences (as for example by Conlisk, 1982, 1992; Bartholomew, 1985; De Palma and Claude Lefevre, 1983, and Kulkarni and Kumar, 1989). In modeling collective population behavior, De Palma and Claude Lefevre (1983) have shown that interactive Markov models follow directly from population-dependent choice behavior exhibited by individuals. With respect to traffic assignment problem in particular, such population dependencies can often be characterized in terms of those agglomeration effects (both positive and negative), which determine the relative attractiveness of a route to road users.

Two stochastic models of the traffic assignment problem are of particular interest -- the logit model (Dial, 1971) and the Probit model (Dagonzo and Sheffi, 1977; Maher and Hughes, 1997). The Probit model is hardly practical because it involves only Monte-Carlo procedures, unless all paths can be identified. The logit model is endowed with both an extremely efficient fixed time assignment procedure (Dial's STOCH algorithm) and a convex minimization formulation with a closed-form objective function (Fisk, 1980). The logit model has attracted much more attention because it has a simple structure and is easy to use (Dial, 1971; Fisk, 1980; Chen and Alfa, 1991; Bell, 1995; Akamatsu, 1996; Leurent, 1997). Nevertheless, computational difficulties have prevented the logit model from being more widely used.

One considerable convenience of the logit model is its Markovian nature (Maher, 1998): the spilt of traffic between paths from an origin to an intermediate point is independent of the spilt between paths from that point to the destination. A Markov chain is a stochastic process that takes on a finite or countable number of possible values. For a Markov chain, the conditional distribution of any future state is independent of the past states and depends only upon the present state. The traditional assumption of a Markov chain is the independence of choice behaviors of individuals in a system in which all transition probabilities are constants. It means that a road user does not consider an interaction with the population in each time period but just independently make his route choice with a constant probability.

Conlisk (1976) introduced an interactive Markov chain that relaxed the assumption of independence; that is, the system status affects an individual's choice behavior. Conlisk defined the transition probabilities as the function of system distribution. Akamatsu (1996) proposed two theoretical approaches for solving the logit-type stochastic traffic assignment model. One is based on the theory of the absorbing Markov process (Markov chain), and the other is based on the equivalency of the maximum entropy principle and the logit model.

Smith and Hsieh (1997) classified those factors that influenced an individual's choice behavior in a system, and introduced the Gravity-type Interactive Markov Models. The individual choice probability is defined as a function of population distribution of the system in each period.

This study proposes a gravity-type interactive Markov model for the stochastic traffic assignment (STA) problem, named GIMT. Within this framework, the daily path choice of drivers in a network can be described, and it is similar to the choice behavior based on the logit model. In this model, the probability that a path be chosen by drivers in each period is a function of the flow distribution in the network. That is, the path choice does depend on the current flow distribution in the network. According to GIMT, we developed an algorithm for finding the SUE of the STA problem. Furthermore, some results of applying the algorithm to two examples are also presented.

This paper is organized as follows. Section 2 briefly describes the GIM model, which underlies both the GIMT model and the GIM algorithm. Section 3 introduces the GIMT model to describe the interactive relationship between path choice and flow distribution in a network. The long-run evolution to the steady state of the process can be observed by interactively adjusting traffic flows among paths. Then, the steady-state conditions of the GIMT model are shown to be equivalent to the SUE conditions in a STA problem. Section 4 develops the GIM algorithm for finding the SUE in a network by determining the steady state of the GIMT model. Section 5 presents two examples to explain the procedure of employing the GIM algorithm to search the SUE of a network. The advantages and disadvantages of the GIM algorithm are illustrated by comparing it to the Frank-Wolfe algorithm and the MSA algorithm. Section 6 provides a summary and some directions for future researches.

2. The Gravity-Type Interactive Markov Models – the GIM Models

2.1 Markov chain

The Markov chain is a stochastic process that describes the evolution of the status of random variables in a system with time (stages). It was first introduced by A. A. Markov in 1907, to model and explain some sociological and economical phenomena. A traditional assumption of the Markov chain is that neither other people nor the distribution of system states affects an individual's choices. The transition probabilities are constant and the individuals do not interact.

For any fixed population of N individuals (*motorists or drivers*) distributed over a finite set of states (*routes or paths*), $i, j \in K = \{1,...,k\}$. $\mathbf{M} = [m_{ij}]$ is a k×k transition matrix. For each individual, the value m_{ij} denotes the conditional probability that a motorist will, when in state i, make a transition into state j. Let $\mathbf{p}^t = (p_1^t, p_2^t, \ldots, p_k^t)$ be the population distribution vector, and p_i^t denotes the population fraction in state i at time t. The population distribution in each period can be represented by a Markov chain with transition matrix, **M**:

(1) $$\mathbf{p}^{t+1} = \mathbf{p}^t \cdot \mathbf{M} \quad \text{or} \quad p_j^{t+1} = \sum_{i=1}^{K} p_i^t \cdot m_{ij} \quad \text{for } i, j \in K,\ t \in \mathbf{Z}_+$$

The transition probability m_{ij} in a traditional Markov chain is assumed to be constant and does not change with time. An ergodic Markov chain will tend to the steady-state probability distribution after a large number of transitions, and this distribution is independent of the initial distribution.

Definition 2-1: Steady State

A *steady state* for a transition matrix, **M**, is a distribution, $\mathbf{p}^* \in \mathbf{P}_k$, which remains invariant in (1), that is, which satisfies the fixed-point condition,

(2) $$\mathbf{p}^* = \mathbf{p}^* \cdot \mathbf{M}$$

The steady-state probability means that the fraction of population in every state keeps constant when the system reaches the steady state. It is important to note that the steady-state probability does not imply that the system settles down into one state. The process continues to make transitions from state to state, and the outflow rate equals the inflow rate for every state.

2.2 Interactive Markov Chains

The assumption of individual independence of Markov chains does not hold in many situations. For example, individual migration decisions will generally depend on the current population sizes in every region and the current popularity of various brands typically influences a consumer's choice of brand.

The class of interactive Markov models first introduced by Matras (1967) and Conlisk (1976) has been widely studied and applied in the social sciences. In this model, each individual's decision as to her or his next state depends only on her or his current state together with the current distribution of all other individuals. It means that individual's decision is influenced by the population distribution. By doing so, the transition probability is defined as a function of the population distribution in a system. That is, **M** is a function of $\mathbf{p}^t$, and can be written as $\mathbf{M}(\mathbf{p}^t)$. Therefore, the interactive Markov chain can be represented in the form as,

$$\text{(3)} \qquad \mathbf{p}^{t+1} = \mathbf{p}^t \cdot \mathbf{M}(\mathbf{p}^t) \qquad \text{and} \qquad p_j^{t+1} = \sum_{i=1}^{K} p_i^t \cdot m_{ij}(\mathbf{P}^t) \quad \text{for } i, j \in \mathrm{K}, \quad t \in \mathbf{Z}_+$$

Several models of the interactive Markov chain were presented in previous researches; the basic one is the A-B model, introduced by Conlisk (1976). Analyzing interactive Markov chains is difficult since the transition probability can be defined in several ways. The formulations are so complex that conditions that support the uniqueness of the steady state are hard to be established. However, these conditions are very important for analyzing a system's structural behavior.

2.3 Gravity-Type Interactive Markov Models – GIM Models

Smith and Hsieh (1994, 1997) introduced a class of interactive Markov migration models which is characterized by gravity-type transition kernels, in which migration flows in each time period are postulated to vary inversely with some symmetric measure of migration costs and directly with some population-dependent measure of attractiveness. This class of models is called gravity-type interactive Markov models.

A gravity-type interactive Markov chain can be briefly expressed as follows. The attractiveness of state j can be written as a function of p_j^t, say $a_j(p_j^t)$. It may then be postulated that $m_{ij}(\mathbf{p}^t)$ increases with $a_j(p_j^t)$. The distance-deterrence effects when moving from i to j are represented by a function, say $g(c_{ij})$, which is defined as a decreasing function of c_{ij}. Then it may also postulated that $m_{ij}(\mathbf{p}^t)$ increases with $g(c_{ij})$. The GIM model can be generalized by allowing **M** to depend on $\mathbf{p}^t$ as follows:

$$\text{(4)} \qquad \mathbf{p}^{t+1} = \mathbf{p}^t \cdot \mathbf{M}(\mathbf{p}^t), \quad t \in \mathbf{Z}_+$$

In (4), the transition probabilities $m_{ij}(\mathbf{p}^t)$ takes the explicit form

$$\text{(5)} \qquad m_{ij}(\mathbf{p}^t) = \frac{a_j(p_j^t)g(c_{ij})}{\sum_k a_k(p_k^t)g(c_{ik})}, \quad i, j \in \mathrm{K}, \quad t \in \mathbf{Z}_+$$

Hence, the transition behavior of a system in each period can be represented as,

(6) $$p_j^{t+1} = \sum_i p_i^t m_{ij}(\mathbf{p}^t) = \sum_i p_i^t \frac{a_j(p_j^t) g(c_{ij})}{\sum_k a_k(p_k^t) g(c_{ik})}, \quad \forall\, i,\ j \in K$$

Smith and Hsieh (1997) established the conditions for the steady state of a GIM model and, showed that there exists a unique steady state for a GIM model if the attraction function $a_j(p_j)$ is strictly decreasing with p_j and the deterrence function $g(c_{ij})$ is symmetric.

3. A GIM Model for Stochastic Traffic Assignment Problem – GIMT Model

By employing an appropriate transformation, we can set up a GIM model for the stochastic traffic assignment problem, which is called the "*GIMT model*". At first, let's recall the following assumptions made in the stochastic traffic assignment (STA) problem.

1. All travel demands on every O-D pair are known.
2. The cost functions on every link are known and separable and the cost function for each link in a congested network is strictly increasing. For example, the U.S.B.P.R.-type volume-delay curves are of the form,

   $t_i = a_i + b_i x_i^4$.
3. The path choice probabilities for motorists are calculated according to multinomial logit model.
4. All paths are non-overlapping, i.e., there is no link in common between two different paths.
5. The notation used herein is as follows.

Consider a transportation network G = (N, A), where

N: set of nodes;

A: set of links (arcs);

R: set of origins, and $R \subseteq N$;

S: set of destinations, and $S \subseteq N$;

K_{rs}: set of paths between r and s, $r \in R$, $s \in S$;

x_a: flow on link a, $\mathbf{x} = (\ldots, x_a, \ldots)$;

t_a: travel time on link a, $\mathbf{t} = (\ldots, t_a, \ldots)$;

f_k^{rs}: traffic flow on path k connecting O-D pair r-s, $\mathbf{f}^{rs} = (\ldots, f_k^{rs}, \ldots)$;

c_k^{rs}: travel time on path k connecting O-D pair r-s, $\mathbf{c}^{rs} = (\ldots, c_k^{rs}, \ldots)$;

q_{rs} : travel demand from origin r to destination s;

$\delta^{rs}_{a,k}$: indicator variable; $\delta^{rs}_{a,k}$ =1, if link a is on path k between O-D pair r-s,

$\delta^{rs}_{a,k}$ =0, otherwise.

Based on these assumptions, the condition for the steady state to be obtained according to a GIM model is equivalent to that for determining the SUE in a stochastic traffic assignment problem. By determining the steady state of the GIMT model, a new algorithm -- the GIM algorithm for determining SUE can be developed.

If the attraction and deterrence functions are of exponential form, then the probability that a driver currently on path i will choose path j in next time period would have the following "*logit*" form:

(7) $$m_{ij}(\mathbf{p}^t)=\frac{\exp\{-\theta[a_j(p_j^t)+c_{ij}]\}}{\sum_k \exp\{-\theta[a_k(p_k^t)+c_{ik}]\}}$$

The link costs increase with link flow, which can depict the effect of congestion in the stochastic traffic assignment problem. Therefore, the transition probability can have a simpler form as

(8) $$m_{ij}(\mathbf{p}^t)=\frac{\exp[-\theta c_{ij}]}{\sum_k \exp[-\theta c_{ik}]}$$

From the viewpoint of behavior, the route choice probability, $m_{ij}(\mathbf{p}^t)$ of drivers in a STA problem can be briefly described as following:

From current flow distribution $\mathbf{p}^t = (\ldots, p_j^t, \ldots)$, we know flows on every path:

$f_j^{rs,t} = p_j^t \cdot q^{rs}$, then we can have flows on each link: $x_a = \sum_{r,s}\sum_k f_k^{rs} \cdot \delta^{rs}_{a,k}$, and hence the travel time on each link: $t_a(x_a)$, and thus we can have the travel time on every path: $c_j^{rs} = \sum_a t_a \cdot \delta^{rs}_{a,j}$, and then obtain the probability that path j is chosen by drivers:

(9) $$m_j(\mathbf{p}^t) = \frac{\exp[-\theta c_j^t]}{\sum_k \exp[-\theta c_k^t]}$$

In this context, the transition probability m_{ij} can be simplified to m_j in the STA problem, because a driver will choose path j in the next time period depends only upon the current flow distribution in the network, and is independent of her or his current state. Therefore, the GIMT model of a STA problem can be represented in the

same form as that of GIM model, $\mathbf{p}^{t+1} = \mathbf{p}^{t} \cdot \mathbf{M}(\mathbf{p}^{t})$.
That is,

(10) $$p_j^{t+1} = \sum_i p_i^t m_j(\mathbf{p}^t) = \sum_i p_i^t \frac{\exp[-\theta c_j^t]}{\sum_k \exp[-\theta c_k^t]}$$

Equations (9) and (10) constitute the GIMT model of a STA problem. If the steady-state condition of the GIMT model can be proven equivalent to the SUE condition in the network, then the SUE can be determined by searching the steady state of the GIMT model.

3.1 Conditions for Stochastic User Equilibrium (SUE) in a network

Given that the travel demand on an O-D pair is q_{rs}, the SUE conditions are (Sheffi, 1985),

(11) $$f_j^{rs} = p_j^{rs} \cdot q^{rs}, \quad \forall \quad j, r, s \text{, or}$$

(12) $$p_j^{rs} = \frac{f_j^{rs}}{q^{rs}}$$

where p_j^{rs} is the probability that route j between r and s is chosen. When the network reaches SUE, p_j^{rs} is just equal to the fraction of the flow on route j.

3.2 The steady-state conditions of the GIMT model are equivalent to the SUE conditions

As defined in Definition 2-1, the steady state probability $\mathbf{p}^*$ of the GIMT model satisfies

(13) $$\mathbf{p}^* = \mathbf{p}^* \mathbf{M}(\mathbf{p}^*)$$

The proportion of the flow on path j in time period t is $p_j^t = \frac{f_j^{rs,t}}{q^{rs}}$ in the GIMT model. Following (10), if all paths do not overlap between r and s, then the proportion of flow on path j in the next period will be

(14) $$p_j^{t+1} = \sum_i p_i^t m_j(\mathbf{p}^t) = \sum_i p_i^t \frac{\exp[-\theta c_j^t]}{\sum_k \exp[-\theta c_k^t]}$$

After a large number of transitions, the system will tend to the steady state. The steady-state probability will satisfy $\mathbf{p}^* = \mathbf{p}^* \mathbf{M}(\mathbf{p}^*)$, and the transition matrix is

(15) $$\mathbf{M}(\mathbf{p}^*) = \begin{bmatrix} m_1 & m_2 & \cdots & m_j \\ m_1 & m_2 & \cdots & m_j \\ \vdots & \vdots & \ddots & \vdots \\ m_1 & m_2 & \cdots & m_j \end{bmatrix}$$

The steady-state probability for any route $j \in K$ satisfies

(16) $$p_j^* = \sum_{i \in K^{rs}} p_i^* \cdot m_j(\mathbf{p}^*), \quad \forall\ j \in K \quad \text{and}$$

$$\sum_i p_i^* = 1 .$$

From (16), we have $p_j^* = \mathrm{m_j}(\mathbf{p}^*)$.

In the meantime, the steady-state conditions can be written as

(17) $$\frac{f_j^{rs*}}{q^{rs}} \equiv p_j^* \quad \Rightarrow f_j^{rs*} = q^{rs} \cdot m_j, \ \ \forall\ j \in K$$

Therefore, the steady-state condition of the GIMT model for a stochastic traffic assignment problem is $\frac{f_j^{rs*}}{q^{rs}} = p_j^* = m_j = p_j^{rs}$, which is same as the SUE condition of the logit-based STA problem, $f_j^{rs} = p_j^{rs} \cdot q^{rs}$.

4. GIM algorithm for finding the SUE of a STA problem

The gravity-type interactive Markov chains describe population-dependent choice behavior by individuals. It is assumed that individual's decision is affected by the population distribution in a system. In STA problems, drivers' transition choices are negatively influenced by the congestion effects on paths, which is quite similar to individual choice behavior in the GIM chains. Therefore, the GIM model would be good to be applied to the STA problem, the choice making process can be represented as follows.

As mentioned before, it is assumed that drivers' path choice is influenced by the flow distribution in a network. The path choice probability in each time period can be defined as a function of the flow distribution over paths. According to these assumptions, the road system will adjust itself to the SUE by iteratively updating the choice probabilities in each time period.

GIMT models can be classified into two categories. For some reasons (e.g. habit), not every driver attempts to change her or his route in each time period. The first type is the “*whole-shifting* GIM model”, in which all drivers are assumed to change their route in each time period. The other is the “*partial-shifting* GIM model” in which only part of drivers (say a proportion $\alpha \in (0,1]$) attempt to change their routes according to the transition matrix in each time period.

4.1 Whole-shifting GIMT model

The whole-shifting GIMT model has the form as (14), in which all individuals are assumed to try to change their state in each time period. The equivalence shown in the preceding section allows the SUE to be obtained by determining the steady state by the following procedure.

Step 1: Initialization. Arbitrarily assign an initial flow f_k^{rs} (n =1) to each path between r-s. Let n denote the number of iteration (or time period). For example, assign equal flows to each path, $f_k^{rs}(1) = \frac{q^{rs}}{K}$, for k =1,…, K. ⇒ The fraction of the flow on path k is $p_k^{rs} = \frac{f_k^{rs}}{q^{rs}}$ ⇒ Calculate the flow on each link by $x_a = \sum_r \sum_s \sum_k f_k^{rs} \delta_{a,k}^{rs}$, yielding $\mathbf{x}^{n=1}$ =(…, x_a^1,…) ⇒ Update the cost on each link according to $t_a^n = t_a^n(x_a)$ ⇒ Calculate the cost on each path by $c_k^{rs}(n) = \sum_a t_a^n(x_a)\delta_{a,k}^{rs}$, for k =1,…, K.

Step 2: Based on the new path cost, $\mathbf{c}^{rs}(n)$ ⇒ Update the choice probabilities for each path using $m_k^{rs}(n) = \frac{e^{-\theta c_k^{rs}(n)}}{\sum_{i \in K_{rs}} e^{-\theta c_i^{rs}(n)}}$ ⇒ Update the fraction of flow on each path using $p_k^{rs}(n+1) = m_k^{rs}(n)$ ⇒ Update the flow on each path with $f_k^{rs}(n+1) = p_k^{rs}(n+1) \times q^{rs}$.

Step 3: Check whether the tolerance of convergence is met. For example, set the tolerance $| f_k^{rs}(n+1) - f_k^{rs}(n) | \le \varepsilon$, $\forall$ r, s, k. If the tolerance is satisfied then STOP, which means that the SUE is found. Otherwise, reset n = n+1, repeat Step 1, 2 and 3.

4.2 Partial-shifting GIMT model

In the real world, not every driver attempts to alters her or his commuting route every day because habit or some other reasons. Assuming only part of drivers (say a proportion,α) attempt to change their routes, and the others to remain their present routes, then the GIMT model becomes

(18) $$p_j^{n+1} = (1-\alpha)\times p_j^n + \sum_i \alpha \times p_i^n \times m_j^n$$

Let us call it the "partial-shifting GIMT model". It is clear that the partial-shifting GIMT model is same as the whole-shifting GIMT model whenα=1. Smith and Hsieh (1997) showed that if α is sufficiently small, then the GIMT model can surely converge on the steady state (which is the SUE of the network).

Multiplying both sides of Eq. (20) by the travel demand q^{rs} , yields

(19) $$f_j^{n+1} = (1-\alpha)\times f_j^n + \alpha\times q^{rs}\times m_j^n$$

Equation (19) states that the flow on route j in next period (n+1) equals the partial remaining flow on j plus that move to j.

Based on Eq. (19), the procedures for implementing the partial-shifting GIMT algorithm are as follows.

Step 1: Initialization. Arbitrarily assign to each path the initial flow f_k^{rs} (n=1). The fraction of flow on path k is $p_k^{rs} = \frac{f_k^{rs}}{q^{rs}}$. ⇒ Calculate the flow on each arc by $x_a = \sum_r \sum_s \sum_k f_k^{rs} \delta_{a,k}^{rs}$, yielding $\mathbf{x}^{n=1}$ ⇒ Update the cost of each arc by $t_a^n = t_a^n(x_a)$ ⇒ Calculate the cost associated with each path with $c_k^{rs}(n) = \sum_a t_a^n(x_a)\delta_{a,k}^{rs}$, $\forall \quad k \in K_{rs}$.

Step 2: Based on the new path cost, $\mathbf{c}^{rs}(n)$, ⇒ Update the choice probability of each path by $m_k^{rs}(n) = \frac{e^{-\theta c_k^{rs}(n)}}{\sum_{i\in K_{rs}} e^{-\theta c_i^{rs}(n)}}$, ⇒ Update the proportion of flow on each path by $p_k^{rs}(n+1) = (1-\alpha)\times p_k^{rs}(n) + \alpha\times m_k^{rs}(n)$, ⇒ Update the flow on each path by $f_k^{rs}(n+1) = p_k^{rs}(n+1)\times q^{rs}$。

Step 3: Check whether the tolerance of convergence is met. For example, set the tolerance $| f_k^{rs}(n+1) - f_k^{rs}(n) | \le \varepsilon$, ∀ r, s, k. If the tolerance is satisfied then

STOP, which means that the SUE is found. Otherwise, reset n = n+1, then return to Step 1 and repeat.

5. Illustrative Examples

Two examples are presented to explain how the GIM algorithm determines the steady state of the corresponding GIMT model. Based on the equivalence shown in Section 3, the steady state determined means that the SUE is obtained. Follows that, the advantages and disadvantages of the GIM algorithm are also discussed.

Example 1: (Paths do not overlap)

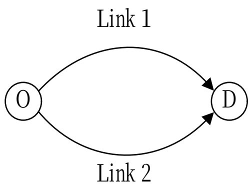

FIGURE 1: Two links (routes) example adopted from Sheffi (1985)

This example is adopted from Sheffi (1985). Consider the simple network shown in Fig. 1. The network includes two links (non-overlapping paths) that connect one O-D pair. The link performance functions are given by,

(20) $$t_1(x_1) = 1.25[1 + (x_1/800)^4]$$

(21) $$t_2(x_2) = 2.50[1 + (x_2/1200)^4]$$

where x_a is measured in vehicles per hour and t_a is measured in minutes. The O-D trip demand is q vehicles per hour (veh/hr).

Given that q = 4000 veh/hr and $\theta = 1.0$ min^{-1}. Solved by using the GIM algorithm, a steady state is obtained after 32 iterations. The convergence pattern of flow on path 1 is shown in Figure 2. When this system reaches its steady state, the flows on path 1 and 2 are x_1 =1781 veh/hr and x_2 =2219 veh/hr, respectively, which are completely satisfy the SUE conditions in Eq. (11). From this point of view, the result is better than the SUE flows x_1 =1845 veh/hr and x_2 =2155 veh/hr, respectively, solved in Sheffi (1985).

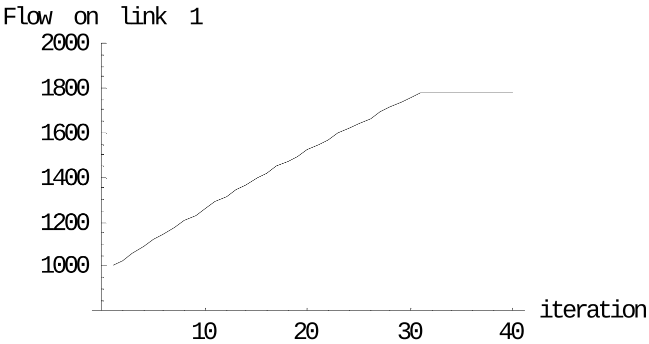

FIGURE 2: Convergence pattern of the GIM algorithm for Example 1; the case of relatively large perception variance (θ =1.0) and relatively high congestion level (q = 4000).

Example 2: (Paths have some arcs in common)

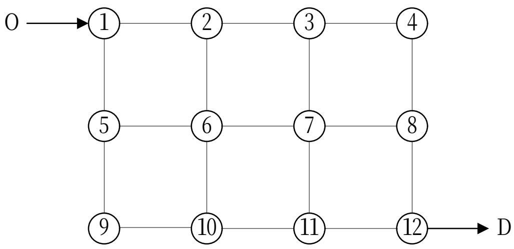

FIGURE 3: 17 links (10 paths) example quoted from Chen and Alfa (1991)

This example is quoted from Chen and Alfa (1991) There are 12 nodes and 17 directed links (horizontally to the right and vertically downward) in the network as shown in Fig. 3. It is used to compare the results solved by GIM algorithm with that of Chen and Alfa. The performance function of each link (i, j) is the BPR curve (i.e., $t(x_{(i,j)}) = a_{(i,j)}+b_{(i,j)} (x_{(i,j)})^4$) and Table 1 presents the corresponding values of $a_{(i,j)}$ and $b_{(i,j)}$. A link (i, j) connects the source node i and the sink node j.

TABLE 1: The value of parameters in Chen and Alfa's example

(i, j)	$a_{(i, j)}$	$b_{(i, j)}$	(i, j)	$a_{(i, j)}$	$b_{(i, j)}$
(1,2)	20.0	0.008	(1,5)	18.0	0.008
(2,3)	23.0	0.008	(2,6)	19.0	0.008
(3,4)	17.0	0.008	(3,7)	16.0	0.008
(4,8)	22.0	0.008	(5,6)	14.0	0.008
(5,9)	24.0	0.008	(6,7)	17.0	0.008
(6,10)	20.0	0.008	(7,8)	13.0	0.008
(7,11)	26.0	0.008	(8,12)	19.0	0.008
(9,10)	7.0	0.008	(10,11)	18.0	0.008
(11,12)	17.0	0.008	--	--	--

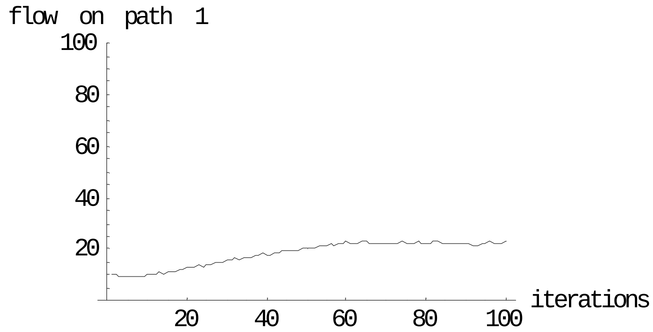

FIGURE 4: Convergence pattern of the flow on path 1

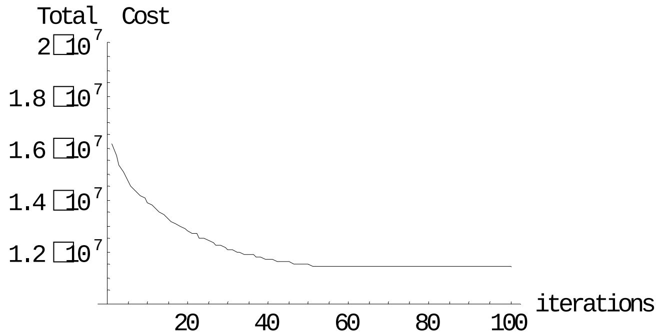

FIGURE 5: Convergence pattern of the total cost

There is only one origin-destination pair in the network, and hence, there are ten paths go from node 1 to node 12. Suppose that the travel demand q = 100 veh/hr. The steady state is determined by using the partial-shifting GIM algorithm with α=0.01. Figure 4 shows the convergence pattern of the flow on path 1(connecting nodes 1-2-3-4-8-12), solved by the software package, MATHEMATICA 4.0. Figure 5 presents the convergence pattern for the total travel cost of the system. The calculation results at iteration #99 are briefly presented as follows (details see in Appendix 2).

Iteration=99
Path Flows: pd1=22.9709; pd2=6.04499; pd3=4.61081; pd4=9.43069; pd5=3.6973
Link Flows: x1=50.452; x2=49.548; x3=33.6267; x4=16.8253; x5=22.9709;
x6=10.6558
Total travel Time=1142547.942
Percentage on Path 1 = 0.229709, Percentage on Path 2 = 0.0604499
Probability for Path 1 = 0.227412, Probability for Path 2 = 0.0598454

The system will converge to the SUE if the shifting proportion is sufficiently small (e.g., α= 0.01), although the convergence speed is quite slow. The computation time needs up to 40 seconds on a personal computer. As shown in Figures 4 and 5, the system clearly converges toward its SUE after 65 iterations. The results of iteration #99 show that the SUE found by the GIM algorithm is very close to the SUE conditions in (11).

Chen and Alfa (1991) solved this problem by improving the MSA algorithm, based on Fisk's optimization model. They ran a Fortran-77 program on a mainframe (AMDAHL-5870) to obtain the SUE. Their result of total travel cost converged to 1.286×10^7, which is obviously higher than the value of 1.145×10^7 solved by the GIM algorithm.

6. Discussions and conclusions

The GIMT model is a new way for analyzing the SUE in a SAT problem. The results of previous examples show that the GIM algorithm is efficient in finding the SUE of a STA problem. In addition, the GIM algorithm can be implemented by a simple program (see Appendix 1 and 2) in MATHEMATICA 4.0 and all the calculations can be completed on a personal computer within few seconds. As regards the method of successive averages (MSA), because of the re-calculation about the flows on each link with STOCH (Dial, 1971) it involves a heavy calculation in each step. In the case of the Frank-Wolfe algorithm, it involves a minimization program for the optimal searching direction and move size in each step (Sheffi & Powell, 1981), heavy calculation work is also involved.

The logit-based choice model assumes that the system exhibits the independence of irrelevant alternatives (IIA). Therefore, no overlapping paths are allowed when the model is applied to solve the stochastic traffic assignment problem. This limitation severely restricts the implementation of the logit-based models. A lot of methods have been proposed to relax the IIA limitation, for example, the nested logit model (Koppelman and Wen, 1998; Wen and Koppelman, 2001; Hensher and Greene, 2002) and the paired combinatorial logit (PCL) model (Koppelman and Wen, 2000). It would be a promising research direction to combine these methods with the GIMT models. It seems that the nested logit model or PCL model can help the GIMT model to deal with a network with overlapping paths.

The GIMT model proposed in this paper is a new approach for analyzing the SUE in a STA problem. In this paper, the steady-state conditions of the GIMT model are shown to be equivalent to the SUE conditions in a STA problem. Within this framework, the dynamics of the daily path choices of commuters in a network can be depicted.

Although the mathematical formulation is similar to the logit model, the GIM algorithm is shown more efficient for finding the SUE in non-overlapping cases than the MSA or the Frank-Wolfe algorithm. In addition, example 2 shows that the GIM algorithm still can converge to the steady state of a GIMT model with a better result than that of Chen and Alfa (1991), even though the IIA property is violated.

References

Akamatsu, T., 1996. "Cyclic Flows, Markov Process and Stochastic Traffic Assignment," *Transportation Research B*, 30(5), 369-386.

Bartholomew, D. J., 1985. "Interactive, Threshold and Non-linear Models for Social Systems," *Journal of Mathematical Sociology*, 11, 43-45.

Bell, M. G. H., 1995. "Alternatives to Dial's Logit Assignment Algorithm," *Transportation Research B*, 29(4), 287-296.

Chen, M. and Alfa, A. S., 1991. "Algorithms for Solving Fisk's Stochastic Traffic Assignment Model," *Transportation Research B*, 25(6), 405-412.

Conlisk, J., 1976. " Interactive Markov Chains," *Journal of Mathematical Sociology*, 4, 157-815.

------. 1982. "The Law of Supply and Demand as a Law of Markov Chains," *Journal of Economic Theory*, 26, 1-16.

------. 1992. "Stability and Monotonicity for Interactive Markov Chains," *Journal of Mathematical Sociology*, 17, 127-143.

Daganzo, C. F. and Sheffi, Y., 1977. "On Stochastic Models of Traffic Assignment," *Transportation Science*, 11(3), 253-274.

De Palma, A. and Claude Lefevre, 1983. "Individual Decision-making in Dynamic Collective Systems," *Journal of Mathematical Sociology*, 9, 103-124.

Dial, R. B., 1971. "A Probabilistic Multipath Traffic Assignment Model which Obviates Path Enumeration," *Transportation Research*, 5, 83-111.

Fisk, C., 1980. "Some Developments in Equilibrium Traffic Assignment Methodology," *Transportation Research B*, 14, 243-256.

Kulkarni, M. S. and Kumar A., 1989. "Interactive Markov Models of 'Progressive' Trends," *Journal of Mathematical Sociology*, 14, 45-65.

Leurent, F. M., 1997. "Curbing the Computational Difficulty of the Logit Equilibrium Assignment Model," *Transportation Research B*, 31(4), 315-326.

Maher, M. J. and Hughes, P. C., 1997. "A Probit-Based Stochastic User Equilibrium Assignment Model," *Transportation Research B*, 31(4), 341-355.

Maher, M., 1998. "Algorithms for Logit-based Stochastic User Equilibrium Assignment," *Transportation Research B*, 32(8), 539-549.

Matras, J., 1967. “Social Mobility and Social Structure: Some Insights from the Linear Model,” *American Sociological Review*, 32, 608-614.

Sheffi, Y., 1985. *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*, New York: Prentice-Hall.

Smith, T. E. and Shang-Hsing Hsieh, 1994, "Gravity-Type Interactive Markov Models：Uniqueness and Stability of Steady States", 41st North American Meetings of RSAI in Canada.

Smith, T. E. and Shang-Hsing Hsieh, 1997 “Gravity-type interactive Markov models – Part I: A programming formulation of steady states”, Journal of Regional Science, 37(4), 653-682.

Smith, T. E. and Shang-Hsing Hsieh, 1997 “Gravity-type interactive Markov models – Part II: Lyapunov stability of steady states”, Journal of Regional Science, 37(4), 683-708.

Appendix 1. The program of example 1

```
k=40
Array[n,k]
r=0.01
td=4000
a=1000
n[1]=a
b=td-a
aq=a/td
bq=b/td

t1[x1_]:=1.25*(1+(x1 / 800)^4)
t2[x2_]:=2.50*(1+(x2 / 1200)^4)

p1[t1_, t2_]:=N[ 1/(1+ Exp[t1 - t2]) ]
p2[t1_, t2_]:=N[ 1/(1+ Exp[t2 - t1]) ]

q1[r_,q_,p_]:=(1-r)*q + r*p
q2[r_,q_,p_]:=(1-r)*q + r*p

iter=1
While[iter<k,
at=t1[a];
bt=t2[b];
```

```
Print["================================="];
Print["iteration #", iter];
Print["x1(", iter, ")=", N[a], ";    x2(", iter, ")=", N[b]];
Print["p1=", N[ap], ";    p2=", N[bp]];
Print["q1=", N[aq], ";    q2=", N[bq]];
Print["t1=", N[at], ";    t2=", N[bt]];
iter=iter + 1;
ap=p1[at, bt];
bp=p2[at, bt];
aq=q1[r,aq,ap];
bq=q2[r,bq,bp];
a=aq * td;
b=bq * td;
n[iter]=a;
]
Print["====================================="]
Print[Array[n,iter]]
ListPlot[ Array[n,iter], PlotRange -> {800,2000},
AxesLabel ->{"iteration", "x1"}]
ListPlot[ Array[n,iter], PlotRange -> {800,2000},
PlotJoined -> True, AxesLabel ->{"iteration", "x1"}]
```

```
===============================
iteration #1
x1[redacted].;  x2[redacted].
p1=ap;  p2=bp
q1=0.25;  q2=0.75
t1=4.30176;  t2=100.156
===============================
…
===============================
iteration #33
x1[redacted].82;  x2[redacted].18
p1=0.423761;  p2=0.576239
q1=0.445205;  q2=0.554795
t1=31.9424;  t2=31.7405
===============================
```

```
iteration #34
x1[redacted].;  x2[redacted].
p1=0.449686;  p2=0.550314
q1=0.44525;  q2=0.55475
t1=31.9548;  t2=31.731
===============================
iteration #35
x1[redacted].96;  x2[redacted].04
p1=0.444296;  p2=0.555704
q1=0.445241;  q2=0.554759
t1=31.9521;  t2=31.733
```

Appendix 2. The program of example 2

```
alpha=0.01;iteration=100;Array[ttt,iteration];Array[tto,iteration]
q=100;fathom=0.0001;tmp=99999;i=0
(* Probability on each Path *)
p1=1/10;     (* Path 1 → 1-2-3-4-8-12 *)
p2=1/10;     (* Path 2 -> 1-2-3-7-8-12 *)
p3=1/10;     (* Path 3 -> 1-2-3-7-11-12 *)
p4=1/10;     (* Path 4 -> 1-2-6-7-8-12 *)
p5=1/10;     (* Path 5 -> 1-2-6-7-11-12 *)
p6=1/10;     (* Path 6 -> 1-2-6-10-11-12 *)
p7=1/10;     (* Path 7 -> 1-5-6-7-8-12 *)
p8=1/10;     (* Path 8 -> 1-5-6-7-11-12 *)
p9=1/10;     (* Path 9 -> 1-5-6-10-11-12 *)
p10=1/10   (* Path 10-> 1-5-9-10-11-12 *)

While[   i<iteration ,
   (* Flow Distribution on each Path pd_ *)
   pd1=q*p1 ;
   pd2=q*p2;
   pd3=q*p3;
   pd4=q*p4;
   pd5=q*p5;
   pd6=q*p6 ;
   pd7=q*p7;
```

```
pd8=q*p8;
pd9=q*p9;
pd10=q*p10;
(* Flow on each Link x_ *)
x1=pd1+pd2+pd3+pd4+pd5+pd6;
x2=pd7+pd8+pd9+pd10;
x3=pd1+pd2+pd3;
x4=pd4+pd5+pd6;
x5=pd1;
x6=pd2+pd3;
x7=pd1;
x8=pd7+pd8+pd9;
x9=pd10;
x10=pd4+pd5+pd7+pd8;
x11=pd6+pd9;
x12=pd2+pd4+pd7;
x13=pd3+pd5+pd8;
x14=pd1+pd2+pd4+pd7;
x15=pd10;
x16=pd6+pd9+pd10;
x17=pd3+pd5+pd6+pd8+pd9+pd10;
(* Travel time(cost) on each Link *)
t1=20.0+0.008*(x1^4);    (* link: 1-2 *)
t2=18.0+0.008*(x2^4);    (* link: 1-5 *)
t3=23.0+0.008*(x3^4);    (* link: 2-3 *)
t4=19.0+0.008*(x4^4);    (* link: 2-6 *)
t5=17.0+0.008*(x5^4);    (* link: 3-4 *)
t6=16.0+0.008*(x6^4);    (* link: 3-7 *)
t7=22.0+0.008*(x7^4);    (* link: 4-8 *)
t8=14.0+0.008*(x8^4);    (* link: 5-6 *)
t9=24.0+0.008*(x9^4);    (* link: 5-9 *)
t10=17.0+0.008*(x10^4);    (* link: 6-7 *)
t11=20.0+0.008*(x11^4);    (* link: 6-10 *)
t12=13.0+0.008*(x12^4);    (* link: 7-8 *)
t13=26.0+0.008*(x13^4);    (* link: 7-11 *)
t14=19.0+0.008*(x14^4);    (* link: 8-12 *)
t15=7.0+0.008*(x15^4);    (* link: 9-10 *)
t16=18.0+0.008*(x16^4);    (* link: 10-11 *)
```

```
t17=17.0+0.008*(x17^4);    (* link: 11-12 *)
(* Travel time(cost) on each Path *)
pt1=t1+t3+t5+t7+t14;
pt2=t1+t3+t6+t12+t14;
pt3=t1+t3+t6+t13+t17;
pt4=t1+t4+t10+t12+t14;
pt5=t1+t4+t10+t13+t17;
pt6=t1+t4+t11+t16+t17;
pt7=t2+t8+t10+t12+t14;
pt8=t2+t8+t10+t13+t17;
pt9=t2+t8+t11+t16+t17;
pt10=t2+t9+t15+t16+t17;
(* The chosen probability on each Path *)
mx=Exp[-pt1]+Exp[-pt2]+Exp[-pt3]+Exp[-pt4]+Exp[-pt5]+Exp[-pt6]+Exp[-pt7]+
      Exp[-pt8]+Exp[-pt9]+Exp[-pt10];
m1=Exp[-pt1]/mx;
m2=Exp[-pt2]/mx;
m3=Exp[-pt3]/mx;
m4=Exp[-pt4]/mx;
m5=Exp[-pt5]/mx;
m6=Exp[-pt6]/mx;
m7=Exp[-pt7]/mx;
m8=Exp[-pt8]/mx;
m9=Exp[-pt9]/mx;
m10=Exp[-pt10]/mx;

ttt[i]=pd1;
(* Note ! The Total Cost means every one spent on their path. *)
tto[i]=(pt1*p1+pt2*p2+pt3*p3+pt4*p4+pt5*p5+pt6*p6+pt7*p7+pt8*p8+pt9*p9+
            pt10*p10)*q;
Print["Iteration=",i];
Print["Path Flows : pd1=",pd1,"; pd2=",pd2,"; pd3=",pd3,
   "; pd4=",pd4, "; pd5=",pd5];
Print["Link Flows: x1=",x1,"; x2=",x2,"; x3=",x3,"; x4=",x4,"; x5=",x5,
   "; x6=",x6];
Print["Total travel Time=",pt1+pt2+pt3+pt4+pt5+pt6+pt7+pt8+pt9+pt10];
i=i+1;
```

```
    p1=(1-alpha)*p1+alpha*m1;
    p2=(1-alpha)*p2+alpha*m2;
    p3=(1-alpha)*p3+alpha*m3;
    p4=(1-alpha)*p4+alpha*m4;
    p5=(1-alpha)*p5+alpha*m5;
    p6=(1-alpha)*p6+alpha*m6;
    p7=(1-alpha)*p7+alpha*m7;
    p8=(1-alpha)*p8+alpha*m8;
    p9=(1-alpha)*p9+alpha*m9;
    p10=(1-alpha)*p10+alpha*m10;

Print["Percentage on Path 1 = ", pd1 / q, "    Percentage on Path 2 = ",
        pd2 / q];
Print["Probability for Path 1 = ", p1, "    Probability for Path 2 = ", p2];
```

Iteration=0
Path Flows : pd1=10; pd2=10; pd3=10; pd4=10; pd5=10
Link Flows: x1=60; x2=40; x3=30; x4=30; x5=10; x6=20
Total travel Time=1612919.0
Percentage on Path 1 = 1/10, Percentage on Path 2 = 1/10
Probability for Path 1 = 0.099, Probability for Path 2 = 0.099
Iteration=1
Path Flows : pd1=9.9; pd2=9.9; pd3=9.9; pd4=9.9; pd5=9.9
Link Flows: x1=59.4; x2=40.6; x3=29.7; x4=29.7; x5=9.9; x6=19.8
Total travel Time=1579454.2212399996
Percentage on Path 1 = 0.099, Percentage on Path 2 = 0.099
Probability for Path 1 = 0.09801, Probability for Path 2 = 0.09801
…..
Iteration=97
Path Flows : pd1=22.417; pd2=6.16772; pd3=4.70443; pd4=9.62217; pd5=3.77237
Link Flows: x1=50.4561; x2=49.5439; x3=33.2892; x4=17.1669; x5=22.417;
x6=10.8722
Total travel Time=1143836.9774758574
Percentage on Path 1 = 0.22417, Percentage on Path 2 = 0.0616772
Probability for Path 1 = 0.221928, Probability for Path 2 = 0.0610605
Iteration=98
Path Flows : pd1=22.1928; pd2=6.10605; pd3=4.65739; pd4=9.52594; pd5=3.73464

Link Flows: x1=49.9515; x2=50.0485; x3=32.9563; x4=16.9952; x5=22.1928; x6=10.7634

Total travel Time=1144083.6063089329

Percentage on Path 1 = 0.221928, Percentage on Path 2 = 0.0610605

Probability for Path 1 = 0.229709, Probability for Path 2 = 0.0604499

Iteration=99

Path Flows : pd1=22.9709; pd2=6.04499; pd3=4.61081; pd4=9.43069; pd5=3.6973

Link Flows: x1=50.452; x2=49.548; x3=33.6267; x4=16.8253; x5=22.9709; x6=10.6558

Total travel Time=1142547.942689541

Percentage on Path 1 = 0.229709, Percentage on Path 2 = 0.0604499

Probability for Path 1 = 0.227412, Probability for Path 2 = 0.0598454