



PII: S0191-2615(96)00003-3

CYCLIC FLOWS, MARKOV PROCESS AND STOCHASTIC
TRAFFIC ASSIGNMENT*

TAKASHI AKAMATSU

Department of Knowledge-based Information Engineering, Toyohashi University of Technology, Toyohashi,
Aichi 441, Japan*(Received 19 January 1995; in revised form 15 November 1995)*

Abstract—Dial's stochastic assignment algorithm restricts the assignment path set to "efficient path." As a result, it sometimes produces the unrealistic flow pattern that no flow is loaded on some paths where many vehicles are running in reality. To remove the drawback of Dial's algorithm, this paper presents the LOGIT type assignment that does not restrict the assignment paths. We first show the theoretical relation between the proposed model and Sasaki's assignment model through Markov process. This analysis makes it clear that the proposed assignment model can be calculated by some matrix operations. Next, we propose an efficient algorithm that does not require the matrix operation nor path enumeration over a network. The algorithm solves an equivalent program based on the entropy decomposition derived from the Markov property of LOGIT model. Finally, it is shown that the proposed approach can be easily extended to the flow dependent case (i.e. stochastic equilibrium assignment). Copyright © 1996 Elsevier Science Ltd

1. INTRODUCTION

The algorithm of Dial (1971) is one of the most popular procedures for the calculations of a LOGIT type stochastic assignment model. The algorithm has an advantage in that it can be easily applied to a large scale network, since it does not require the path enumeration over a network, and its computational burden is almost same order as the minimum path assignment. The Dial's algorithm however, sometimes produces an unrealistic flow pattern in that no flow is loaded on some paths where many vehicles are running in reality, since the algorithm restricts the path set for loading flows.

To remove this drawback of Dial's algorithm, this paper presents the LOGIT type stochastic assignment that does not restrict the assignment paths[†]. Two theoretical approaches for solving the model are also shown: 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. The former approach results in the procedure recently developed by Bell (1995), where he did not explore the theoretical basis. In the latter approach, we construct the equivalent mathematical program for the model, which is solved by using the decomposition of entropy function [for details of the decomposition, see Akamatsu (1990, 1994) and Akamatsu and Tsuchiya (1990). Before Bell (1995) proposed the model, Akamatsu (1990) had already studied the same model using the latter approach].

This paper is organized as follows: the next section begins with the explanations on the reasons why Dial's algorithm restricts the assignment paths. Then, the stochastic assignment model that removed the path restriction is introduced. In Section 3, we explore

*Chapter 3 of this paper was originally written as a comment on the paper by Bell (1995). Other parts of this paper are *partial* summaries of the author's doctor dissertation submitted to Tokyo University (March, 1990).

[†] Some researchers have pointed out the drawbacks such as IIA property in the LOGIT type assignment (Florian and Fox, 1976; Daganzo and Sheffi, 1977). In order to overcome the IIA property, the PROBIT type assignment model (Daganzo and Sheffi, 1977; Daganzo, 1982) has been proposed. It is hard, however, for the model to be applied for large scale networks because of the heavy computational burden required for the assignment or the parameter estimation. Moreover, there are many cases where the biases arising from the IIA property of the LOGIT model do not cause serious problems when we consider the flow dependent assignment (i.e. stochastic equilibrium assignment). Thus, the LOGIT type assignment model remains important, and the improvement of the model is meaningful from the viewpoint of practical applications.

the calculation method of the proposed assignment model using the absorbing Markov chain theory: it is first demonstrated that the proposed model has some theoretical relationships to the assignment model through Markov process (Sasaki, 1965). Moreover, the precise procedure for solving the model is derived. In Section 4, we show another approach that is applicable to a large scale network: we first construct an equivalent mathematical program. Next, an efficient algorithm for solving the program is developed. The approach uses the decomposition scheme of the entropy function derived from the Markov property of the LOGIT model. Furthermore, some results of numerical experiments of the algorithm are shown. Finally, it is discussed that the latter approach can be easily extended to the flow-dependent case (i.e. stochastic equilibrium assignment).

2. STOCHASTIC ASSIGNMENT WITH CYCLIC FLOWS

2.1. Dial's algorithm and cyclic flows

In Dial's algorithm, the paths for loading flows are restricted to the 'efficient path' which is defined as the path including only links that take the traveler further away from the origin. More specifically, the each link in the path must increase the minimum cost from the origin. As a result, it sometimes produces an unrealistic flow pattern. For a concrete example, consider the network shown in Fig. 1. This is a simplified network of Japanese Metropolitan Expressway. All the links in this network are always used by many vehicles in reality. When we consider the path between origin node o and destination node d , there are two paths: $o \rightarrow 1 \rightarrow 6 \rightarrow 5 \rightarrow d$ and $o \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow d$. Dial's algorithm, however, does not permit one to choose the path: $o \rightarrow 1 \rightarrow 6 \rightarrow 5 \rightarrow d$. The reason is that this path is not 'efficient' in the sense of Dial's definition: the minimum cost for node 5 from the origin is $4+2+2+3=11$, while the minimum cost for node 6 is 12, which means that going from node 6 to node 5 does not take the traveler away from the origin.

The main reason why Dial's algorithm restricts the assignment paths seems to originate from avoiding the occurrence of 'cyclic flows' in the procedures. Note that 'cyclic flows' means not only the case that flows are loaded on an explicit cycle but also the case that the flows on multiple paths form apparent cycles (loops). To illustrate this, consider the

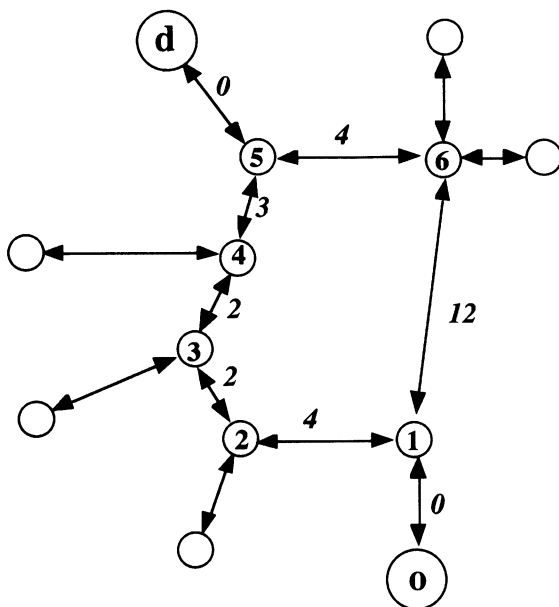


Fig. 1. Japanese Metropolitan Expressway (the numbers written beside the links denote the link travel times).

network shown in Fig. 2. Suppose that the path set is defined as

$$\{1:1 \rightarrow 2 \rightarrow 4, 2:1 \rightarrow 3 \rightarrow 4, 3:1 \rightarrow 2 \rightarrow 3 \rightarrow 4, 4:1 \rightarrow 3 \rightarrow 2 \rightarrow 4\}.$$

The flows on links $2 \rightarrow 3$ and $3 \rightarrow 2$ exist when the flows on paths 3 and 4 are not zero. Consequently, when we see only the resulting link flow pattern, the cyclic flows which loops between nodes 2 and 3 can exist, whereas the definition of the path set does not contain the cycle. Alternatively, if we consider the path set as in Dial's definition, cyclic flows cannot arise, since paths 3 and 4 are excluded.

The reason why Dial's algorithm avoids the cyclic flows can be easily understood by tracing the procedure. The 'forward pass' of the algorithm *successively* calculates the conditional probability of choosing a link ('link weight'). The result depends on the order of calculation, since the values of link-weight for processed links are used for the evaluation in the subsequent links. Consequently, it is not possible to decide the order of the *sequential* calculation which causes no contradiction in the evaluation procedure, if the link set (i.e. implicit path set) for assignment contains the cyclic flow. Furthermore, a cyclic flow causes the similar contradictions in the 'backward pass' of the algorithm. Thus, we can see that Dial's algorithm carefully avoids the cyclic flow to maintain the consistency of the *sequential* procedure, and that the restriction of the path set to 'efficient paths' is the derivative result.

2.2. LOGIT type assignment with infinite cycles

We are seeking the stochastic assignment method which does not avoid an 'inefficient path'. The natural definition of the assignment path for this purpose is 'simple path': the path which does not pass through same link more than once. This conventional definition, however, has the possibility of generating apparent 'cyclic flows', which means that the sequential algorithm such as Dial's algorithm cannot be applied. Furthermore, there is a possibility that the definition generates such long detours as real users do not consider them in a large scale network. Therefore, it seems to be difficult for this definition to develop the efficient algorithms for solving the stochastic assignment in a large scale network.

Thus, in contrast to Dial's strategy of restricting the assignment paths, we take the strategy that does not restrict the paths for the assignment. That is, we define the path set which contains all the paths including infinite cycles as the assignment path set. Although this definition may look rather strange from the viewpoint of reasonable user behavior, the justification of this definition can be made from the viewpoint of the computational feasibility. As shown in the later sections, this definition enables us to implement the stochastic assignment in very large scale networks. Moreover, the definition is useful in some iterative algorithms for solving the stochastic equilibrium assignment. It is well known that the change of path set due to the change of link cost pattern during the iterations causes the oscillatory behavior of the algorithm (Van Vliet and Dow, 1979; Powell and Sheffi, 1982; Davis, 1994). Since the path set based on the definition above does not vary, regardless of the link cost pattern, it can avoid this troublesome problem. Thus, we think that it is valuable to try the new definition and to investigate the properties. Note, also, that the definition is same as the implicit one in Sasaki's assignment model through the Markov process (Sasaki, 1965), which has been applied in various practical networks.

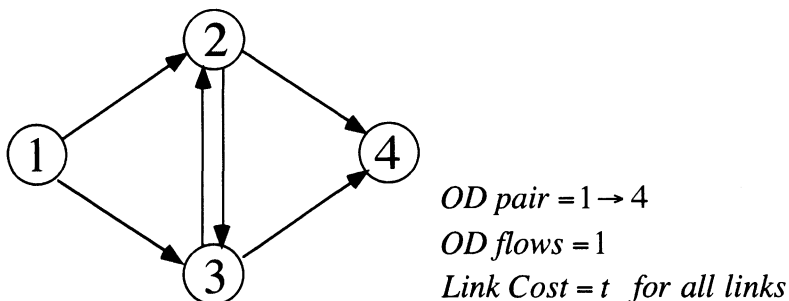


Fig. 2. Example network.

To illustrate the difference between these definitions of path set, consider a network shown in Fig. 2. The assignment path set by Dial's definition is given as follows*:

$$\begin{aligned} 1: & 1 \rightarrow 2 \rightarrow 4 \\ 2: & 1 \rightarrow 3 \rightarrow 4 \end{aligned}$$

In the conventional definition (i.e. simple path), the following paths are added to the assignment path set:

$$\begin{aligned} 3: & 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \\ 4: & 1 \rightarrow 3 \rightarrow 2 \rightarrow 4 \end{aligned}$$

On the other hand, our new definition permits us to include the following infinite cycles into the path set in addition to the conventional paths:

$$\begin{aligned} & 1 \rightarrow 2 \rightarrow 3 \rightarrow 2 \rightarrow 4, 1 \rightarrow 2 \rightarrow 3 \rightarrow 2 \rightarrow 3 \rightarrow 4, \dots, 1 \rightarrow 2 \rightarrow 3 \rightarrow \dots \rightarrow 2 \rightarrow 3 \rightarrow 4 \\ & 1 \rightarrow 3 \rightarrow 2 \rightarrow 3 \rightarrow 4, 1 \rightarrow 3 \rightarrow 2 \rightarrow 3 \rightarrow 2 \rightarrow 4, \dots, 1 \rightarrow 3 \rightarrow 2 \rightarrow \dots \rightarrow 3 \rightarrow 2 \rightarrow 4 \end{aligned}$$

As we change the definition of path set, the probabilities of path choice also vary. When we assume the LOGIT model for path choice, the probabilities of choosing paths in each definition are given as follows:

(a) Dial's definition:

$$P_1 = P_2 = \frac{1}{2}, \tag{1}$$

where P_r : the probability that the r th path is chosen,

(b) Conventional definition:

$$P_1 = P_2 = \frac{1}{2(1 + A)}, \quad P_3 = P_4 = \frac{A}{2(1 + A)}, \tag{2}$$

where $A \equiv \exp(-\theta t)$, θ : the dispersion parameter for LOGIT model,

(c) New definition (i.e. definition without path restriction):

$$P_r = \frac{\exp[-\theta C_r]}{\sum_{r=1}^{\infty} \exp[-\theta C_r]} \tag{3a}$$

$$= \frac{\exp[-\theta C_r]}{2(A^2 + A^3 + A^4 + \dots)} = \frac{\exp[-\theta C_r]}{\{2A^2/(1 - A)\}}, \tag{3b}$$

where C_r : the travel cost for the r th path.

Note that the path cost in (3) is defined as follows:

$$C_r = \sum_{ij} t_{ij} \delta_{ij,r}, \tag{4}$$

where $\delta_{ij,r} = \begin{cases} n & \text{if the } r\text{th path passes through link } ij \text{ } n \text{ times} \\ 0 & \text{otherwise} \end{cases}$

Equation (4) and the definition of $\delta_{ij,r}$ mean that if a certain path (cycle) passes through a certain link n times, we should count the n times of the link cost in evaluating the path cost.

The link flows can be calculated by appropriately summing the path flows. In conventional assignment models, the probabilities (flows) of choosing links are related with the probabilities of path choice by the following equations:

$$P_{ij} = \sum_r P_r \hat{\delta}_{ij,r}, \tag{5}$$

where $\hat{\delta}_{ij,r} = \begin{cases} 1 & \text{if the } r\text{th path passes through link } ij \\ 0 & \text{otherwise} \end{cases}$

*For the illustrative purpose, we adopted here the criterion of strictly increasing the distance from the origin.

Hence, the link flows for each definition in the example network are given by

(a) Dial's definition:

$$p_{12} = p_{13} = p_{24} = p_{34} = \frac{1}{2}, \quad p_{23} = p_{32} = 0, \quad (6)$$

(b) Conventional definition:

$$p_{12} = p_{13} = p_{24} = p_{34} = \frac{1}{2}, \quad p_{23} = p_{32} = \frac{A}{2(1+A)}, \quad (7)$$

where p_{ij} , flow (probability) on link $i \rightarrow j$.

On the other hand, when we consider cyclic flows, the path flows and link flows should be linked, not with eqn (5), but with the following relationship:

$$p_{ij} = \sum_{r=1}^{\infty} P_r \delta_{ij,r}, \quad (8)$$

The definition of $\delta_{ij,r}$ is the same as that in eqn (4). Hence, eqn (8) means that, if a certain path (cycle) passes through a certain link n times, we should count the n times of the path flows in measuring the link flow. Thus, substituting the path-choice probability in eqn (3b) into eqn (8), we obtain the following link flows for the definition without path restriction in the example network.

(c) New definition (i.e. definition without path restriction):

$$p_{12} = p_{13} = p_{24} = p_{34} = \frac{1}{2},$$

$$p_{23} = p_{32} = \frac{A}{2(1-A)}. \quad (9)$$

In summary, the stochastic assignment model considered in this paper consists of the following equations, which is different from the conventional LOGIT type assignment in the definition of the assignment path set and the incidence matrix:

(1) Path-choice probability according to the LOGIT model:

$$P_r^{od} = \frac{\exp[-\theta C_r^{od}]}{\sum_{r=1}^{\infty} \exp[-\theta C_r^{od}]},$$

where P_r^{od} : the probability that the r th path between nodes o and d is chosen,

(2) Relation between path cost and link cost:

$$C_r^{od} = \sum_{ij} t_{ij} \delta_{ij,r},$$

where C_r^{od} : the travel costs for the r th path between nodes o and d ,

$$\delta_{ij,r} = \begin{cases} n & \text{if the } r\text{th path between nodes } o \text{ and } d \text{ passes through link } ij \text{ } n \text{ times} \\ 0 & \text{otherwise} \end{cases}$$

(3) Relation between link flow, path flow and OD flow:

$$p_{ij}^{od} = \sum_{r=1}^{\infty} P_r^{od} \delta_{ij,r}^{od},$$

$$x_{ij}^{od} = q_{od} p_{ij}^{od}.$$

where p_{ij}^{od} : the probability that link ij is chosen for the trips with OD – pair od ,
 x_{ij}^{od} : the flow on link ij with OD – pair od ,
 q_{od} : the flow for OD – pair od .

3. ABSORBING MARKOV PROCESS AND LOGIT ASSIGNMENT

Dial’s algorithm indicates that the stochastic assignment based on LOGIT model has the property of the Markov process [the Markov property in LOGIT type assignment is also discussed in Akamatsu (1990, 1996), where some interesting properties are derived]. On the other hand, the definition of path set (including infinite cycles) introduced in the previous chapter is same as that in Sasaki’s assignment model through the absorbing Markov process (we call it MPA). Considering these facts, we can expect that the LOGIT-type assignment with infinite cycles has some theoretical relationships with the MPA. Thus, this chapter examines the relation between these models, which will lead us to the derivation of the calculation procedure for the LOGIT type assignment.

3.1. Traffic assignment through the absorbing Markov process

We will begin with briefly reviewing the basic idea of the MPA. In MPA, the nodes in a network are considered as the ‘states’ in the Markov process. The movements of vehicles between two incident nodes in a network correspond to the ‘events’ (transition of the states) in the Markov process (Markov chain). It is assumed that the vehicles generated from an origin node repeat changing their states according to the Markov process rule. When they move to the state corresponding to a destination node, they are absorbed with a probability of 1.

To give a formal representation of the MPA, consider a network with n nodes whose g elements are origins, and a elements are destinations. The probabilistic movement of vehicles in MPA is fully described by the following matrix whose elements denote the transition probability (conditional probability) between two nodes:

$$P = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{R} & \mathbf{Q} \end{bmatrix} \begin{matrix} a \\ n-a \end{matrix} \tag{10}$$

where \mathbf{I} denotes a unit matrix, the element of matrix \mathbf{R} means the transition probability from a node excluding of destinations to a destination node, and the element of matrix \mathbf{Q} means the transition probability between a node pair excluding of destinations. Note that the transition probability is zero when the corresponding node pair is not linked with a link (i.e. the nodes reached from a node by one step movements are restricted to the incident nodes). In addition, to distinguish origin nodes from traversal nodes, the matrix \mathbf{Q} is assumed to be arranged in the following order:

$$Q = \begin{bmatrix} \mathbf{0} & \mathbf{Q}_1 \\ \mathbf{0} & \mathbf{Q}_2 \end{bmatrix} \begin{matrix} g \\ n-g-a \end{matrix} \tag{11}$$

According to the Markov chain rule, the probability that the vehicle stayed at node i at an initial time will be staying at node j after n transitions (i.e. after passing n links) is given by the (i, j) element of the matrix Q^n . Since the vehicles arrive at each node through various transition steps (i.e. various paths including cycles), the probabilities that the vehicle generated from origins pass through the nodes over a network are calculated by $\mathbf{I} + Q^1 + Q^2 + \dots$. If we can assume that the matrix \mathbf{Q} satisfies the Hawkins–Simon condition [see, for example, Takayama (1985)], i.e. if all the successive principal minors of $(\mathbf{I} - \mathbf{Q})$ are positive, the matrix series converges, and the sum is given by the following inverse matrix:

$$\begin{aligned} \mathbf{I} + Q^1 + Q^2 + \dots &= [\mathbf{I} - \mathbf{Q}]^{-1} \\ &= \begin{bmatrix} \mathbf{I} & \mathbf{Q}_1[\mathbf{I} - \mathbf{Q}_2]^{-1} \\ \mathbf{0} & [\mathbf{I} - \mathbf{Q}_2]^{-1} \end{bmatrix} \begin{matrix} g \\ n-g-a \end{matrix} \end{aligned} \tag{12}$$

The matrix $\mathbf{Q}_1 [\mathbf{I} - \mathbf{Q}_2]^{-1}$ on the right-hand side of (12) is meaningful for a traffic assignment, since it gives the node-choice probabilities for each origin node. That is, the (i, j) element of the matrix gives a probability that a vehicle generated from the i th origin passes through node j . The node-choice probability multiplied by the transition probability yields the link-choice probability (link flow), which is the required output for a traffic assignment.

When applied to the network shown in Fig. 2, the transition probability matrix takes the following form:

$$\mathbf{P} = \begin{bmatrix} 4 & 1 & 2 & 3 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & p(2|1) & p(3|1) \\ p(4|2) & 0 & 0 & p(3|2) \\ p(4|3) & 0 & p(2|3) & 0 \end{bmatrix} \begin{matrix} 4 \\ 1 \\ 2 \\ 3 \end{matrix} \quad (13)$$

Substituting this into eqn (12), the probabilities of choosing nodes 2 and 3 are given by

$$\begin{aligned} [P(2), P(3)] &= \mathbf{Q}_1 [\mathbf{I} - \mathbf{Q}_2]^{-1} \\ &= \frac{1}{d} [p(2|1), p(3|1)] \begin{bmatrix} 1 & p(3|2) \\ p(2|3) & 1 \end{bmatrix} \\ &= \frac{1}{d} [p(2|1) + p(3|1) \cdot p(2|3), p(3|1) + p(2|1) \cdot p(3|2)], \end{aligned} \quad (14)$$

where $d \equiv 1 - p(2|3) \cdot p(3|2)$.

The link-choice probabilities are obtained by multiplying the above node-choice probabilities and the transition probabilities, i.e.

$$p_{ij} = P(i) \cdot p(j|i). \quad (15)$$

As seen in the explanations above, the MPA calculates the link flows from eqn (12), which does not operate path variables explicitly. It is worth noting, however, that the model gives the probability of choosing the r th path from node o to destination d by the equation:

$$P_r^{od} = \prod_{ij} p(i|j)^{\delta_{ij,r}^{od}}, \quad (16)$$

and the link flows are given by the sum of the infinite path flows [eqn (8)].

3.2. Transition probabilities consistent with the LOGIT model

In Sasaki's original MPA model, the transition probability matrix \mathbf{P} is assumed to be determined from the observed link flows. Consequently, the model has been considered to have no background for the driver's route-choice behavior. The MPA, however, can be the behavioral model based on the random utility theory, if the value of \mathbf{P} is set so as to have a proper relation with travel costs. In particular, the MPA is equivalent to the LOGIT type assignment presented in the previous chapter when the transition probabilities (conditional probabilities) are given by the following formula:

$$p(j|i) = \exp[-\theta(t_{ij} + S_{jd} - S_{id})] = \exp[-\theta t_{ij}] \frac{V_{jd}}{V_{id}}, \quad (17)$$

where C_r^{id} : the travel cost of the r th path from node i to destination d ,
 t_{ij} : travel cost of link $i \rightarrow j$,

$$V_{id} \equiv \sum_{r=1}^{\infty} \exp[-\theta C_r^{id}], \quad (18a)$$

$$S_{id} \equiv -\frac{1}{\theta} \ln \sum_{r=1}^{\infty} \exp[-\theta C_r^{id}] = -\frac{1}{\theta} \ln V_{id}. \quad (18b)$$

Clearly, the ‘probabilities’ given by eqn (17) are always non-negative, and satisfy the condition:

$$\sum_j p(j|i) = \frac{\sum_j \exp[-\theta t_{ij}] V_{jd}}{V_{id}} = \frac{V_{id}}{V_{id}} = 1, \tag{19}$$

which is required to be regarded as the conditional probabilities.

The equivalence can be easily understood, since the path-choice probability in MPA yields the following expression:

$$\begin{aligned} \prod_{ij} p(j|i)^{\delta_{ij,r}^{od}} &= \prod_{ij} \exp[-\theta (t_{ij} + S_{jd} - S_{id}) \delta_{ij,r}^{od}] \\ &= \exp[-\theta \sum_{ij} t_{ij} \delta_{ij,r}^{od}] \exp[-\theta \sum_{ij} (S_{jd} - S_{id}) \delta_{ij,r}^{od}] \\ &= \exp[-\theta C_r^{od}] \exp[-\theta (S_{dd} - S_{od})] \\ &= \frac{\exp[-\theta C_r^{od}]}{\sum_{r=1}^{\infty} \exp[-\theta C_r^{od}]} \end{aligned} \tag{20}$$

For a concrete example of this equivalency, consider the network shown in Fig. 2. Evaluating eqn (18a) in this network, we have

$$V_{14} = \frac{2A^2}{1-A}, \quad V_{24} = V_{34} = \frac{A}{1-A}, \quad V_{44} = 1. \tag{21}$$

Hence, eqn (17) gives the following transition probabilities:

$$\begin{cases} p(2|1) = p(3|1) = 1/2 \\ p(3|2) = A, \quad p(4|2) = 1 - A \\ p(2|3) = A, \quad p(4|3) = 1 - A. \end{cases} \tag{22}$$

Substituting these transition probabilities into eqn (14), we obtain the node-choice probabilities:

$$[P(2), P(3)] = \frac{1}{1-A^2} \left[\frac{1}{2}(1+A), \frac{1}{2}(1+A) \right] = \left[\frac{1}{2(1-A)}, \frac{1}{2(1-A)} \right]. \tag{23}$$

The node-choice probability multiplied by the transition probability yields the link-choice probability as follows:

$$\begin{aligned} p_{12} &= 1 \cdot p(2|1) = \frac{1}{2}, & p_{13} &= 1 \cdot p(3|1) = \frac{1}{2}, \\ p_{23} &= P(2) \cdot p(3|2) = \frac{A}{2(1-A)}, & p_{24} &= P(2) \cdot p(4|2) = \frac{1}{2}, \\ p_{32} &= P(3) \cdot p(2|3) = \frac{A}{2(1-A)}, & p_{34} &= P(3) \cdot p(4|3) = \frac{1}{2}. \end{aligned} \tag{24}$$

This is the same result as the LOGIT type assignment shown in Section 2 [eqn (9)].

It may be worth a mention in passing that eqn (17) is applicable to a many-to-one OD pattern. This is assured by the fact that (17) includes no subscripts for origin nodes. Concerning this property, we should be careful with the definition of the conditional probability. In eqn (17), the conditional probability is defined ‘forward’ (see Fig. 3(a)):

$$p(j|i) = \frac{P_{ij}}{P(i)} = \frac{P_{ij}}{\sum_m P_{im}}. \tag{25}$$

In contrast to the definition above, when we define the conditional probability ‘backward’ (see Fig. 3(b)),

$$p(i|j) = \frac{P_{ij}}{P(j)} = \frac{P_{ij}}{\sum_m P_{mj}}, \tag{26}$$

eqn (17) should be replaced with

$$p(i|j) = \exp[-\theta(t_{ij} + S_{oi} - S_{oj})] = \exp[-\theta t_{ij}] \frac{V_{oi}}{V_{oj}}, \tag{27}$$

where C_r^{oi} : the travel cost of the r th path from origin o to node i ,

$$S_{oi} \equiv -\frac{1}{\theta} \ln \sum_{r=1}^{\infty} \exp[-\theta C_r^{oi}] = -\frac{1}{\theta} \ln V_{oi}, \tag{28a}$$

$$V_{oi} \equiv \sum_{r=1}^{\infty} \exp[-\theta C_r^{oi}]. \tag{28b}$$

Then, the formula is applicable to a one-to-many OD pattern.

3.3. Calculations of infinite path costs and link flows

The evaluation method of \mathbf{V} in eqn (18) or eqn (28) needs considerations, since the definition contains the sum of variables related to infinite paths, which means the impossibility of naïve calculation. Recently, an efficient method of the evaluation has been proposed by Bell (1995), though the relation between \mathbf{P} and \mathbf{V} is not explicitly discussed in his paper. The method is similar to the idea of eqn (12) in MPA, which is described as follows.

Consider a matrix \mathbf{W} with n rows and n columns, whose $[i, j]$ element is given as follows:

$$w_{ij} = \begin{cases} \exp[-\theta t_{ij}] & \text{if node pair } (i, j) \text{ is connected by a link} \\ 0 & \text{otherwise.} \end{cases} \tag{29}$$

The typical element of the squares of this matrix yields

$$\begin{aligned} w_{ij}^{[2]} &= \sum_{k=1}^n w_{ik} w_{kj} \\ &= \sum_{k=1}^n \exp[-\theta(t_{ik} + t_{kj})] \\ &= \sum_{r \in R_n^{ij}} \exp[-\theta C_{r,2}^{ij}], \end{aligned} \tag{30}$$

where R_n^{ij} : the set of paths which connect nodes i and j by passing through n links,

$C_{r,n}^{ij}$: the cost of the r th path belonging to R_n^{ij} .

Similarly, the typical element of \mathbf{W}^n is given by

$$w_{ij}^{[n]} = \sum_{r \in R_n^{ij}} \exp[-\theta C_{r,n}^{ij}]. \tag{31}$$

Hence, we can see that evaluating eqn (18a) is equivalent to the sum of matrix series:

$$\mathbf{V} = \mathbf{W} + \mathbf{W}^2 + \mathbf{W}^3 + \dots, \tag{32}$$

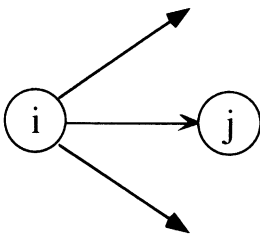


Fig. 3(a) Forward conditional probability.

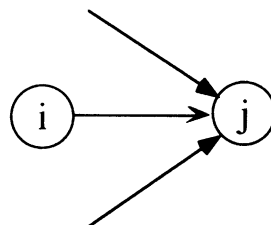


Fig. 3(b) Backward conditional probability.

where the typical element of matrix \mathbf{V} is

$$V_{ij} = \sum_{r=1}^{\infty} \exp[-\theta C_r^{ij}]. \tag{33}$$

Furthermore, the following formula holds for the matrix \mathbf{W} :

$$\mathbf{I} + \mathbf{W} + \mathbf{W}^2 + \mathbf{W}^3 + \dots = [\mathbf{I} - \mathbf{W}]^{-1}, \tag{34}$$

if the matrix \mathbf{W} satisfies the Hawkins–Simon condition. Thus, we can obtain \mathbf{V}_{ij} by the following matrix operations:

$$\mathbf{V} = [\mathbf{I} - \mathbf{W}]^{-1} - \mathbf{I}. \tag{35}$$

Substituting this into eqn (17) or eqn (27) gives the transition probabilities \mathbf{Q}_1 and \mathbf{Q}_2 consistent with LOGIT-type assignment with infinite cycles.

Note that whether the matrix \mathbf{W} satisfies the Hawkins–Simon condition or not is related to the value of dispersion parameter θ . If the value is very low (i.e. users are likely to choose routes with cycles), cyclic flows possibly become extremely large, and the resulting link flow pattern becomes infinity. Therefore, for the convergence of this assignment model, the θ should have the lower bound that depends on the network structure.

The analyses so far show that the straightforward method to achieve the LOGIT-type assignment with infinite cycles requires the following two steps:

- (a) the calculation of the transition probabilities, and
- (b) the calculation of the link-choice probabilities.

Step (a) corresponds to the ‘forward pass’ in Dial’s algorithm. This requires the calculation of \mathbf{V} which reduces to the calculations of eqns (35) and (17). On the other hand, step (b) corresponding to the ‘backward pass’ reduces to the calculation of eqn (12) according to the Markov process theory. Therefore, the entire calculation requires the two inverse matrix calculations: $[\mathbf{I} - \mathbf{W}]^{-1}$ and $\mathbf{Q}_1 [\mathbf{I} - \mathbf{Q}_2]^{-1}$. The consideration below, however, shows that the single inverse matrix is enough: the calculation of $\mathbf{Q}_1 [\mathbf{I} - \mathbf{Q}_2]^{-1}$ reduces to some simple calculations, which do not require additional calculations of inverse matrices.

Given a network with a many to one OD pattern, suppose the matrix \mathbf{W} is arranged in the following order:

$$\mathbf{W} = \begin{bmatrix} & 1 & g & n & g-1 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & & \\ \mathbf{W}_{d1} & \mathbf{0} & \mathbf{W}_1 & & \\ \mathbf{W}_{d2} & \mathbf{0} & \mathbf{W}_2 & & \end{bmatrix} \begin{matrix} 1 \\ g \\ n-g+1 \end{matrix}, \tag{36}$$

where the first column/row and the next g columns/rows correspond to a destination node and origin nodes, respectively. Then, the inverse matrix is given by

$$[\mathbf{I} - \mathbf{W}]^{-1} = \begin{bmatrix} & 1 & g & n & g-1 \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & & \\ \mathbf{V}_{d1} & \mathbf{I} & \mathbf{V}_1 & & \\ \mathbf{V}_{d2} & \mathbf{0} & \mathbf{V}_2 & & \end{bmatrix} \begin{matrix} 1 \\ g \\ n-g+1 \end{matrix}, \tag{37}$$

where

$$\begin{aligned} \mathbf{V}_{d1} &= \mathbf{W}_{d1} + \mathbf{W}_1[\mathbf{I} - \mathbf{W}_2]^{-1}\mathbf{W}_{d2}, \\ \mathbf{V}_{d2} &= [\mathbf{I} - \mathbf{W}_2]^{-1}\mathbf{W}_{d2}, \\ \mathbf{V}_1 &= \mathbf{W}_1[\mathbf{I} - \mathbf{W}_2]^{-1}, \\ \mathbf{V}_2 &= [\mathbf{I} - \mathbf{W}_2]^{-1}. \end{aligned} \tag{38}$$

Since we set the transition probabilities by eqn (17), the transition probability vectors \mathbf{Q}_1 and \mathbf{Q}_2 in eqn (11) are represented as follows:

$$\begin{aligned} \mathbf{Q}_1 &= \bar{\mathbf{V}}_{d1}^{-1} \mathbf{W}_1 \bar{\mathbf{V}}_{d2}, \\ \mathbf{Q}_2 &= \bar{\mathbf{V}}_{d2}^{-1} \mathbf{W}_2 \bar{\mathbf{V}}_{d2}, \end{aligned} \quad (39)$$

where $\bar{\mathbf{V}}_{d1}$ is the $g \times g$ diagonal matrix whose diagonals are \mathbf{V}_{d1} ,
 $\bar{\mathbf{V}}_{d2}$ is the $(n - g - 1) \times (n - g - 1)$ diagonal matrix whose diagonals are \mathbf{V}_{d2} .

Furthermore, substituting eqn (39) into eqn (12) in MPA, the node-choice probability yields

$$\begin{aligned} \mathbf{Q}_1[\mathbf{I} - \mathbf{Q}_2]^{-1} &= \bar{\mathbf{V}}_{d1}^{-1} \mathbf{W}_1 \bar{\mathbf{V}}_{d2} [\mathbf{I} - \bar{\mathbf{V}}_{d2}^{-1} \mathbf{W}_2 \bar{\mathbf{V}}_{d2}]^{-1} \\ &= \bar{\mathbf{V}}_{d1}^{-1} \mathbf{W}_1 \bar{\mathbf{V}}_{d2} \bar{\mathbf{V}}_{d2}^{-1} [\mathbf{I} - \mathbf{W}_2]^{-1} \bar{\mathbf{V}}_{d2} \\ &= \bar{\mathbf{V}}_{d1}^{-1} \mathbf{W}_1 [\mathbf{I} - \mathbf{W}_2]^{-1} \bar{\mathbf{V}}_{d2} \\ &= \bar{\mathbf{V}}_{d1}^{-1} \mathbf{V}_1 \bar{\mathbf{V}}_{d2}. \end{aligned} \quad (40)$$

The typical term of eqn (40) is written as follows:

$$P(i)^{od} = V_{oi} V_{id} / V_{od}. \quad (41)$$

Hence, the link-choice probability is

$$p_{ij}^{od} = p(j|i) V_{oi} V_{id} / V_{od}, \quad (42)$$

where p_{ij}^{od} is the link-choice probability for the vehicle with origin o and destination d . Substituting eqn (17) into eqn (42), we have

$$p_{ij}^{od} = V_{oi} w_{ij} V_{jd} / V_{od}. \quad (43)$$

Thus, we see that the inverse matrix calculation required for the LOGIT assignment is only $[\mathbf{I} - \mathbf{W}_2]^{-1}$ in eqn (38).

Note that equation (43) is the same formula which was originally found by Van Vliet (1981) for the analysis of Dial's algorithm. Although Bell (1995) also suggested the use of the same formula for the LOGIT assignment with infinite cycles, the justification he gave for so doing was informal. The validity has been proved by the above discussions. We must not forget, however, that the proof is valid only for the LOGIT assignment with infinite cycles or the Dial's algorithm (i.e. LOGIT assignment with efficient paths). In other words, the validity of using eqn (43) is not guaranteed for the model with the other definition of path set. Therefore, Bell's "first logit assignment method" (where the other definition of path set is adopted) can not assure us that it yields assignment patterns consistent with LOGIT model (i.e. proving the consistency of the method is an open problem), while his "second logit assignment method" gives correctly the LOGIT assignment pattern with infinite cycles.

4. ENTROPY MODEL AND THE LOGIT ASSIGNMENT

We have seen so far that solving the LOGIT type assignment with infinite cycles reduces to the inverse matrix operations of eqn (38) and the assignment eqn (43). The calculation, however, may be difficult for the large scale network with some thousands of nodes, since it requires extremely vast storages. Thus, this section presents another approach to solve the assignment problem for the large scale network: we construct an equivalent program, and develop an algorithm for solving it.

4.1. Entropy decomposition and cyclic flows

Akamatsu (1989, 1990, 1994) showed that, if the stochastic assignment has the Markov property:

$$P_r = \prod_{ij} p(i|j)^{\delta_{ijr}}, \quad (44a)$$

where

$$p(i|j) = p_{ij} / \sum_m p_{mj}, \quad (44b)$$

then the decomposition of the entropy function:

$$-\sum_r P_r \ln P_r = -\sum_{ij} p_{ij} \ln p_{ij} + \sum_j \left(\sum_i p_{ij} \right) \ln \left(\sum_i p_{ij} \right) \quad (45)$$

holds. The decomposition eqn (45) also holds for the LOGIT type assignment, since the model satisfies eqn (44).

In using eqn (45), we should be careful with the definition of path set. The decomposition does not necessarily hold under the conventional definition of path set (i.e. simple path) and eqn (5) (the relationship between link flows and path flows). For example, when applied to the network shown in Fig. 2, the evaluations of the entropy functions for the probabilities according to the model [i.e. eqns (2) and (7)] are as follows:

$$\begin{aligned} -\sum_{r=1}^4 P_r \ln P_r &= \ln 2 + \ln(1+A) - \frac{A}{1+A} \ln A, \\ -\sum_{ij} p_{ij} \ln p_{ij} &= \frac{1}{1+A} \{ (2+3A) \ln 2 + A \ln(1+A) - A \ln A \}, \\ -\sum_j \left(\sum_i p_{ij} \right) \ln \left(\sum_i p_{ij} \right) &= \frac{1+2A}{1+A} \{ \ln 2 + \ln(1+A) - \ln(1+2A) \}. \end{aligned} \quad (46)$$

Clearly, the decomposition scheme (45) does *not* hold. The reason is that the definition of the path set and the equation for the path-choice probability (44) contradicts each other: that is, if apparent cyclic flows exist, the choice probabilities of the paths containing the cycles are not zero from eqn (44), while such paths are not included in the path set from the definition.

On the other hand, eqn (45) always holds for the assignment with infinite cycles, since the path-choice probability [eqn (44)] and the definition of the infinite path set do not contradict each other. For example, according to this model, the probabilities for the network shown in Fig. 2 are given by eqns (3b) and (9). Substituting them into eqn (45), we have

$$\begin{aligned} -\sum_{r=1}^{\infty} P_r \ln P_r &= \ln 2 - \ln(1-A) - \frac{A}{1-A} \ln A, \\ -\sum_{ij} p_{ij} \ln p_{ij} &= \frac{1}{1-A} \{ (2-A) \ln 2 + A \ln(1-A) - A \ln A \}, \\ -\sum_j \left(\sum_i p_{ij} \right) \ln \left(\sum_i p_{ij} \right) &= \frac{1}{1-A} \{ \ln 2 + \ln(1-A) \} \end{aligned} \quad (47)$$

We see that eqn (45) does hold for this definition.

4.2. An equivalent program with the LOGIT assignment

Consider the mathematical programming:

[P-PATH]

$$\text{Min. } Z(\mathbf{x}(\mathbf{f})) = \sum_{ij} x_{ij} t_{ij} + \frac{1}{\theta} \sum_{od} \sum_r f_r^{od} \ln f_r^{od} \quad (48)$$

subject to

$$\sum_r f_r^{od} = q_{od}$$

$$f_r^{od} \geq 0$$

$$x_{ij} = \sum_{od} \sum_r f_r^{od} \delta_{ij,r}^{od},$$

where f_r^{od} = flow on path r with origin o and destination d ,
 q_{od} = flow from origin o to destination d ,
 x_{ij} = flow on link ij ,
 t_{ij} = travel cost for link ij .

This is the equivalent program for the LOGIT type assignment as shown in Fisk (1980). Although Fisk's original program did not consider the infinite cycles, let us define the path set for the program above so as to include infinite cycles. Then, it is clear that the program is equivalent to the LOGIT-type assignment with infinite cycles.

Since the decomposition holds for the definition of path set including the infinite cycles as seen in Section 4.1, let us substitute eqn (45) into the objective function of [P-PATH]. In addition, replace the flow conservation of arc-chain form with the arc-node representation. Then, we obtain the following program represented by only link variables:

[P-LINK]

$$\text{Min. } Z(\mathbf{x}) = \sum_{ij} x_{ij} t_{ij} + \frac{1}{\theta} \sum_o \{-HL(\mathbf{x}^o) + HN(\mathbf{x}^o)\} \quad (49)$$

subject to

$$\begin{aligned} \sum_i x_{ik}^o - \sum_j x_{kj}^o + q_{od} \delta_{o,k} - \sum_d q_{od} \delta_{d,k} &= 0 \\ \sum_o x_{ij}^o &= x_{ij} \\ x_{ik}^o &\geq 0 \end{aligned}$$

where $HL(\mathbf{x}^o) = - \sum_{ij} x_{ij}^o \ln x_{ij}^o$,

$$HN(\mathbf{x}^o) = \sum_j \left(\sum_i x_{ij}^o \right) \ln \left(\sum_i x_{ij}^o \right),$$

x_{ij}^o = flow on link ij with origin o ,
 δ_{ab} = Kronecker's delta.

This program is also equivalent to the LOGIT type assignment with infinite cycles [Exactly speaking, the derivation above does not necessarily mean the equivalency. For the formal proof, see Akamatsu (1990, 1996).] Therefore, we can obtain the flow pattern for the assignment with infinite cycles by solving the program.

4.3. An algorithm for solving the equivalent program

[P-LINK] is a convex programming that has a unique solution [see Akamatsu (1990), Akamatsu and Tsuchiya (1990)]. Moreover, the program does not contain the path variables. Therefore, it can be solved without path enumerations by applying the standard algorithms for the convex multi-commodity network programming. When we apply the Frank–Wolfe method, the algorithm for this program is described as follows:

Step 1: set an iteration counter $n = 1$, and set a feasible solution x^n .

Step 2: revise the 'link cost':

$$\bar{t}_{ij}^{o,n} \equiv t_{ij} + \frac{1}{\theta} \ln r_{ij}^{o,n} \quad (50)$$

where $x_{ij}^{o,n}$ = flow on link ij with origin o at iteration n

$$r_{ij}^{o,n} \equiv x_{ij}^{o,n} / \sum_i x_{ij}^{o,n}$$

Step 3: solve the following program for each origin o :

[LP-LINK]

$$\min_{\mathbf{y}^o} \sum_{ij} \bar{t}_{ij}^o \cdot y_{ij}^o \tag{51}$$

subject to

$$\sum_i y_{ik}^o - \sum_j y_{kj}^o + q_{od} \delta_{o,k} - \sum_d q_{od} \delta_{d,k} = 0$$

$$y_{ik}^o \geq 0$$

Step 4: solve the line search problem:

$$\min_{\alpha} Z(\mathbf{x}^n + \alpha(\mathbf{y} - \mathbf{x}^n)) \quad \text{s.t. } 0 \leq \alpha \leq 1$$

Revise the temporary solution:

$$x_{ij}^{o,n+1} = x_{ij}^{o,n} + \alpha(y_{ij}^o - x_{ij}^{o,n}) \quad \forall ij, o$$

Step 5: terminate if $\mathbf{x}^{n+1} \approx \mathbf{x}^n$.

Otherwise, increase the iteration number: $n = n + 1$, and go to step 2.

In step 1, various methods are available for finding a feasible solution. For example, Dial’s algorithm gives a good initial solution. The ‘link cost’ in step 2, \bar{t}_{ij}^o , is the coefficient for the linearized term for the objective function of [P-LINK]. The linear program [LP-LINK] reduces to a minimum cost assignment problem over a network with link cost $\{\bar{t}_{ij}^o\}$. Note, however, that the value of “link cost”, \bar{t}_{ij}^o , can be negative. This causes the following two problems in solving the program [LP-LINK]:

(1) The minimum path search algorithm should work correctly, even if the negative link costs are included in a network.

(2) There is a possibility that the negative cycles exist, i.e. the sum of the link costs in the cycle can be negative. When the negative cycle exists, the solution of the program [LP-LINK] becomes infinite.

For the first problem, we should apply a label correcting method for the minimum path search. The label-setting method, such as Dijkstra’s algorithm, does not work for the negative link cost. For the second problem, we propose the following procedure that gives the auxiliary flow vector \mathbf{y}^o such that $\mathbf{y}^o - \mathbf{x}^{o,n}$ yields the feasible descent vector of [P-LINK]: For each origin o :

Step 2-1: Construct the set Γ_o whose elements are the links included in the negative cycles.

Step 2-2: Construct the set Λ_{od} whose elements are the links included in a proper path connecting origin o and destination d .

Step 2-3: For each destination d , load OD flow q_{od} to the links $ij \in \Lambda_{od}$. Set the resulting link flow pattern \mathbf{y}^o .

Step 2-4: Add the flow Q to the links $ij \in \Gamma_o$, where Q is determined by the following equations;

$$\begin{cases} Q = \rho + \gamma & \text{if } \gamma > 0 \\ Q = \rho & \text{if } \gamma \leq 0, \end{cases} \tag{52a}$$

where

$$\gamma \equiv \frac{\sum_{ij \in L} \bar{t}_{ij}^o x_{ij}^{o,n} - \sum_{ij \in \Lambda_o} \bar{t}_{ij}^o y_{ij}^o}{\sum_{ij \in \Gamma_o} \bar{t}_{ij}^o}, \tag{52b}$$

where ρ is a non-negative real valued scalar, and $A_o = \cup_d A_{od}$.

In step 2-1, including only a few cycles into Γ_o is enough for ensuring the convergence, while it is desirable to make all the cycles include into Γ_o from the view point of the fast convergence. In steps 2-2 and 2-3, we have some freedom for selecting the paths for assignment. For example, the minimum cost assignment or the Dial's assignment over a network with 'true link costs', t_{ij} , are available. Equation (52) in step 2-4 ensures that the direction vector $y^o - x^o$ becomes the descent vector of the objective function, i.e.

$$\nabla Z(x^n) \cdot (y - x^n) < 0 \tag{53}$$

holds. Hence, the sequence of the flow pattern generated by this algorithm always decreases the value of objective function. Furthermore, since the flow pattern generated in each iteration is always feasible, the convergence to the rigorous solution is guaranteed.

4.4. Numerical example

We implemented some numerical experiments of the algorithm in the network shown in Fig. 2. In the implementation of the algorithm, we used the minimum cost assignment for finding the initial feasible solution. In steps 2-2 and 2-3 of the algorithm, we first construct the set of links excluding of Γ_o . Then, the minimum cost (with respect to "link cost" \bar{t}_{ij}^o) assignment over the link set was used.

Table 1 depicts the convergence pattern of the link flow, where the dispersion parameter $\theta = 1$, OD flow $q = 1$ and the loading quantity parameter ρ required in step 2-4 is set $10 q$. We can see that the algorithm makes a sequence of flow pattern that converges to the rigorous solution. It should be note that faster convergence could be achieved by using Dial's algorithm for finding the initial solution.

To see the effect of the dispersion parameter to the convergence, we experimented for the various values of θ . The cyclic flow (flow on link 2→3) for each case is 4.75416 ($\theta = 0.1$), 0.29098 ($\theta = 1.0$), 0.00002 ($\theta = 10.0$), respectively. From the value of the cyclic flow, we judged the range of the dispersion parameter ($0.1 \leq \theta \leq 10.0$) realistic. Table 2 shows the number of iterations required for the various convergence criteria ϵ and the various values of θ , where the following stopping rule is used:

$$\text{if } \epsilon > \max_{ij} \left| \frac{x_{ij}^{n+1} - x_{ij}^n}{x_{ij}^n} \right|, \text{ then stop.}$$

The result shows that the algorithm stops within a moderate number of iterations even for generating a very accurate solution, such as $\epsilon = 0.1\%$.

Furthermore, we investigated the effect of the loading quantity parameter ρ in steps 2-4. Table 3 shows the number of iterations for the various values of ρ , where ϵ is set to be 1%. In almost all cases, the algorithm converged within a moderate number of iterations. When $\theta=0.1$ and $\rho=1.0$, however, the algorithm requires too many iterations. It seems that we should set a large value for ρ for the fast convergence when θ is very small (i.e. when very large cyclic flows exist).

Table 1. Convergence pattern ($\theta = 1, q = 1, \rho = 10$)

Iteration no.	Link 1→2	1→3	2→3	3→2	2→4	3→4
1	1.0000	0.0000	0.0000	0.0000	1.0000	0.0000
2	0.9665	0.0335	0.0335	0.0335	0.9665	0.0335
3	0.5406	0.4594	0.1871	0.1871	0.5406	0.4594
4	0.5347	0.4653	0.2930	0.2930	0.5347	0.4653
5	0.5062	0.4938	0.2773	0.2773	0.5062	0.4938
6	0.5054	0.4946	0.2913	0.2913	0.5054	0.4946
7	0.5010	0.4990	0.2888	0.2888	0.5010	0.4990
8	0.5009	0.4991	0.2910	0.2910	0.5009	0.4991
9	0.5002	0.4998	0.2906	0.2906	0.5002	0.4998
10	0.5001	0.4999	0.2909	0.2909	0.5001	0.4999
Rigorous solution	0.5000	0.5000	0.2909	0.2909	0.5000	0.5000

Table 2. Number of iterations required for the convergence ($\rho = 10$)

ϵ	θ		
	0.1	1.0	10.0
5%	3	6	5
1%	4	7	6
0.5%	5	9	6
0.1%	6	11	6

Table 3. Number of iterations required for the convergence ($\epsilon = 1\%$)

ρ	θ		
	0.1	1.0	10.0
1.0	153	8	4
10.0	4	7	6
100.0	9	9	6
1000.0	9	9	4

5. DISCUSSIONS

This paper presented the LOGIT type stochastic assignment which does not restrict the assignment paths. Two approaches for solving the model were provided: one was based on the theory of the Absorbing Markov Process (Markov Chain), and the other was based on the equivalency of the maximum entropy problem and the LOGIT model. Through these analyses, the theoretical linkage between some assignment models (i.e. Markov Process Assignment, LOGIT Assignment with infinite paths and Decomposed Entropy Model) were shown.

Although this model overcomes the path restriction in Dial's algorithm, we should be careful with the potential weaknesses of the model as mentioned below.

First, there is a possibility that the link flows in this model become infinity [i.e. (32) does not converge], due to the cyclic flows. This can occur when the value of θ is very low, and the network for the assignment contains several loops with very small costs. It is, however, plausible to think that the assignment under the appropriate valued parameter (θ with moderately high value) will converge in many applications for real road networks. The reason for the conjecture is that Sasaki's Markov Process Assignment has been applied for some real road networks, and no reports for the non-convergence have been made as far as the author knows. Nevertheless, since we have no empirical results for this model itself at present, we should not consider that this model can be substituted completely for Dial's algorithm; or rather, this model should be the "relief" method only for the case that the Dial's path restriction rule obviously yields the impractical flow pattern.

The second potential weakness is that this model can exacerbate the problem arising from IIA property of LOGIT assignment model. The typical case can be seen in the same example that de Ortúzar and Willumsen (1990) showed for the IIA weakness of Dial's assignment. The bias, however, can be diminished, to some extent, when the model is extended to the flow-dependent case (i.e. stochastic equilibrium assignment), since the over-loading on some links due to the IIA property increases the costs in the over-loaded links, and then it will decrease the flows due to the equilibrium principle. Whether the resulting equilibrium flow patterns are acceptable for practical applications or not depends on the specific conditions applied, and, therefore, the extensive numerical (empirical) examinations are needed for the definite conclusion of this problem.

The extension to the stochastic equilibrium assignment can be achieved easily by the entropy model approach presented in this paper, though we concentrated only on the analyses for the flow-independent case so far. For the flow-dependent case, the modification of the equivalent program is only the first term of the objective function: the total cost, $x_{ij}t_{ij}$, is replaced with the integral of the link cost function $\int_0^{x_{ij}} t_{ij}(\omega) d\omega$. Therefore, the required modification of the algorithm is just one point: The 'link cost' which is used in solving the auxiliary program [LP-LINK] should be replaced with the following form:

$$t_{ij}(x_{ij}^n) + \frac{1}{\theta} \ln r_{ij}^{o,n},$$

where $r_{ij}^{o,n} \equiv x_{ij}^{o,n} / \sum_i x_{ij}^{o,n}$

The remaining steps of the algorithm are all the same with one for the fixed cost network. Although the modification of the procedure is very simple, the judgment for the efficiency of the algorithm needs more extensive numerical experiments. The precise investigation will be reported in a subsequent paper.

Acknowledgements—The author is indebted to Yuji Tsuchiya for extensive and illuminating discussions of these ideas. The author wishes to thank Michael G. H. Bell and anonymous referees who pointed out the weakness of the model discussed in Section 5. The author also wishes to thank Masao Kuwahara for many suggestions which improved the presentation of this paper.

REFERENCES

Akamatsu T. and Matsumoto Y. (1989) A stochastic user equilibrium model with elastic demand and its solution method (in Japanese), *Proceedings of JSCE* **401**, pp. 109–118.

Akamatsu T. (1990) Integrated transportation network models based on stochastic equilibrium approach (in Japanese), Dr course Dissertation, Department of Civil Engineering, Tokyo University, Tokyo.

Akamatsu T. and Tsuchiya Y. (1990) Parallel distributed processing on neural network for some transportation equilibrium assignment problems, In *Proceedings of the 11th International Symposium on the Theory of Traffic Flow and Transportation*, Yokohama (Koshi M., ed. pp. 307–323). Elsevier, New York.

Akamatsu T. (1996) Decomposition of path choice entropy in general transport networks, *Trnsn Sci*. In press.

Bell M. G. H. (1995) Alternatives to Dial's LOGIT Assignment Algorithm. *Trnsn Res.* **29B**, 287–296.

Carre B. (1979) *Graphs and Networks*. Clarendon Press, Oxford.

Daganzo C. F. (1982) Unconstrained external formulation of some transportation equilibrium problems. *Trnsn Sci.* **16**, 332–360.

Daganzo C. F. and Sheffi Y. (1977) On stochastic models of traffic assignment. *Trnsn Sci.* **11**, 253–274.

Davis G. A. (1994) Exact local solution of the continuous network design problem via stochastic user equilibrium assignment, *Trnsn Res.* **28B**, 61–75

Dial R. B. (1971) A probabilistic multipath traffic assignment algorithm which obviates path enumeration. *Trnsn Res.* **5**, 83–111.

Fisk C. S. (1980) Some developments in equilibrium traffic assignment. *Trnsn Res.* **14B**, 243–255.

Florian M. and Fox B. (1976) On the probabilistic origin of Dial's Multipath Traffic Assignment Model. *Trnsn Res.* **10**, 339–341.

Janson B. N. and Zozaya-Gorostiza C. (1987) The problem of cyclic flows in traffic assignment. *Trnsn Res.* **21B**, 299–310.

LeBlanc L. J., Morlok E. and Pierskella W. (1975) An efficient approach to solving the road network equilibrium assignment problem. *Trnsn Res.* **9**, 309–318.

Ortúzar J. de D. and Willumsen L. G. (1990) *Modeling Transport*. John Wiley, West Sussex.

Powell W. B. and Sheffi Y. (1982) The convergence of equilibrium algorithm with predetermined step sizes. *Trnsn Sci.* **16**, 45–55.

Sasaki T. (1965) Theory of traffic assignment through Absorbing Markov Process (in Japanese). *Proc. JSCE* **121**, 28–32.

Sheffi Y and Powell W. B. (1982) An algorithm for the equilibrium assignment problem with random link times. *Networks* **12**, 191–207.

Takayama A. (1985) *Mathematical Economics*. Cambridge University Press, Cambridge.

Van Vliet D. (1981) Selected node-pair analysis in Dial's assignment algorithm. *Trnsn Res* **15B**, 66–68.

Van Vliet D and Dow P. (1979) Capacity-restrained road assignment. *Traffic Engng Control* June 296–305.

APPENDIX: ANOTHER PROOF FOR THE VALIDITY OF (43)

Substituting eqn (17) into eqn (43), we have

$$p_{ij}^{od} = p(j|i)V_{oi}V_{id}/V_{od}. \tag{A1}$$

The term related to **V** in eqn (a1) is rewritten as follows:

$$\begin{aligned} \frac{V_{oi}V_{id}}{V_{od}} &= \frac{\sum_{r=1}^{\infty} \exp[-\theta C_r^{oi}] \sum_{r=1}^{\infty} \exp[-\theta C_r^{id}]}{\sum_{r=1}^{\infty} \exp[-\theta C_r^{od}]} \\ &= \sum_{r' \in R(oid)} \frac{\exp[-\theta C_{r'}^{od}]}{\sum_{r=1}^{\infty} \exp[-\theta C_r^{od}]}, \end{aligned} \tag{A2}$$

where $R(oid)$ is the set of paths which pass through node i and connects origin o and destination d .

The right-hand side of eqn (a2) is the sum of the probabilities of choosing paths which pass through the node i in LOGIT-type assignment with infinite cycles. That is, eqn (a2) means the probability that node i is chosen, $P(i)^{od}$. Consequently, eqn (a1) yields the following equation:

$$p_{ij}^{od} = p(j|i) \cdot P(i)^{od}, \quad (\text{A3})$$

which means the validity of eqn (43) for giving the link-choice probability consistent with the LOGIT assignment with infinite cycles.