Revised July 3, 1996
submitted to *Transportation Science* (*July 1, 1996*)

Stochastic Traffic Assignment with Geometric Attributes of Paths

Takashi Akamatsu

Department of Knowledge-based Information Engineering,
Toyohashi University of Technology, Toyohashi, Aichi 441, Japan
E-mail: akamatsu@tutkie.tut.ac.jp

The purpose of this paper is to develop the LOGIT type stochastic (equilibrium) assignment that satisfies three requirements: first, the path set for the loading should be defined by the simple paths without any restrictions, which do not depend on the travel cost pattern; secondly, the model should give small probabilities for the paths with mazy structure, while the number of paths defined by the first requirement itself is enormous; and finally, the model should be efficiently computable even in large scale networks. To achieve the purpose, we employ two strategies: first, we incorporate not only the conventional travel times but also the geometric attributes of paths into the model; and secondly, we construct the algorithm making fully use of the Markov property of LOGIT model. The incorporation of the geometric aspects into the model is not only natural to satisfy the second requirement but also effectively utilized to achieve the first requirement. Moreover, the strategy combined with the Markov chain theory enables the model to inherit the computational efficiency of Markov Chain Assignment (MCA) developed by ***SASAKI*** *(1965),* ***BELL*** *(1995) and* ***AKAMATSU*** *(1996a). This property stems from the fact that the new model can be regarded as a mathematically natural extension of MCA: the conventional MCA reduces to a system of linear equations in real number space; and the new assignment model yields the almost same form of linear equations in complex number space. Finally, an algorithm for the flow dependent case (i.e. stochastic equilibrium case) is also presented by extending the results for the flow independent model.*

How we should define the path set for the traffic assignment is an open problem yet to be studied despite the long research history. In deterministic (Wardrop) equilibrium model, we do not have to explicitly realize the problem, since the deterministic model assigns the flows only to the shortest paths. However, when once we realize the fact that not only the shortest paths but also the other various paths are chosen by users in real traffic networks, we can not escape from this problem. To date, various stochastic (equilibrium) assignment models (e.g. **SASAKI** (1965), **VON FALKENHAUSEN** (1966), **BURRELL** (1967), **DIAL** (1971), **TOBIN** (1977), **DAGANZO** and

Sheffi (1977), **Fisk** (1980), **Sheffi** and **Daganzo** (1980), **Daganzo** (1982,1983), **Mirchandani** and **Soroush** (1987), **Akamatsu** (1989,1990, 1996a), **Bell** (1995), etc.) have tackled this problem. These models, either explicitly or implicitly, decide the set of paths for assigning flows by a priori criteria, and then the route-choice probabilities are calculated over the path set.

Conventionally, some criteria for defining the path set have been employed. The simple and natural one is to define all the simple paths (i.e. the paths that do not traverse any links more than once), P_S, as the path set for the assignment. To the author's knowledge, however, no efficient algorithm for generating the flow pattern that completely satisfies this definition has been developed. This difficulty is mainly caused by the fact that the path enumeration is computationally impossible in real large-scale networks: the required storage and operations in enumerating simple paths increase exponentially with the growth of the network size. Note that the essential problem can not be *theoretically* overcome even if we utilize the conventional techniques such as column generation, simplicial decomposition, or Monte Carlo simulation.

In order to avoid the path enumeration, some models / algorithms take the strategy that restricts the path set for the assignment to a certain subset of P_S. For example, **Dial** (1971) developed an efficient algorithm which generates the link flow pattern being consistent with the LOGIT type route choice model over the set of *"efficient path"*. Although the path restriction strategy is useful for the efficient calculation, it gives rise to the problem that the algorithm often generates unrealistic flow pattern (for the typical example, see **Akamatsu** (1996a)). The restriction of paths also causes another troublesome problem in the flow dependent assignment: the path set for the loading varies with the change of the link cost pattern, and as a result, all the iterative algorithms in which the Dial's algorithm is utilized can not be guaranteed to converge (for further detail, see **Akamatsu** (1996b,1996c)). Recently, **Laurent** (1996) proposed the modified definition of the efficient path, where the paths for the loading are restricted by some criteria based on the fixed "reference travel costs". Although the definition has the advantage of stabilizing the paths in the flow dependent assignment, it suffers from over-restriction of paths: for example, in the ring-road network presented in Akamatsu(1996a), the definition can produce the unrealistic flow pattern as in the Dial's algorithm.

Recently, **Bell** (1995) and **Akamatsu** (1996a) showed a definition being in a striking contrast to the restriction strategy: they analyzed the LOGIT assignment whose path set consists of all the possible paths, where even paths with cycles are permitted. The model overcomes the deficiency of the Dial's algorithm, and is applicable to large scale networks, since it avoid path enumeration by applying the Markov chain theory (Henceforth, we call the model MCA: Markov Chain Assignment). The definition of the path set, however, is unnatural from the user's behavior point of view, and therefore, it leaves rooms for various improvement. Thus, we can conclude that there is no assignment model equipped with both the behaviorally satisfactory path set and the computationally efficient algorithm.

This study aims to develop the efficient method for obtaining the flow pattern according to the LOGIT based stochastic (equilibrium) assignment model whose path set consists of only simple paths without any restrictions. To achieve the purpose, we employ two strategies: first, we incorporate not only the conventional travel times but also the spatial / geometric attributes of paths into the model; and secondly, we construct the algorithm exploiting the Markov property of the LOGIT model.

The first strategy, obviously, can be justified from the user's behavior view point. Our model due to this strategy assigns only a small amount of flow to the paths with excessively mazy structure even if the travel times of the paths are moderate. This is consistent with the observations in various traffic surveys. Interestingly, the incorporation of the geometric attributes not only improves the behavior model but also gives us the innovative method for constructing the set of simple paths. To be specific, we first define the geometric "rotation-angle" between adjacent links over the network, and then, by representing the rotations-angles along paths in *complex number space*, we obtain the method for distinguishing the cycles from the simple paths without explicit enumeration of paths. This method plays an important roll in developing the efficient assignment algorithm based on the second strategy.

The second strategy implies that the assignment algorithm can avoid explicitly dealing with vast path variables. Namely, our algorithm operates only link variables instead of path variables, and it generates the link flows by origins / destinations. Note that this does *not* mean that our algorithm can produce less information than path based algorithms. From the Markov property of the LOGIT assignment, the link flows by origin / destination give us enough information to construct the corresponding path flows: we can obtain any path flows from the output of this algorithm "as wee needed". To exploit this property, we construct the assignment algorithm based on the MCA developed by **SASAKI**(1965), **BELL**(1995) and **AKAMATSU**(1996a). Although the original MCA assigns the flows over the path set with cycles, the MCA combined with the first strategy can successfully eliminate all the cycles from the path set. Thus, we can achieve the purpose.

The paper is organized as follows. We first briefly review the MCA that is consistent with LOGIT based route-choice model in Section 1, where we also draw attention to the problem caused by the cycle flows in MCA. With this problem in mind, the subsequent sections develop the methods for reducing / eliminating the cycle flows in MCA. In Section 2, we improve the MCA so as to avoid the simplest cycle flows of "U-turn": we modify the MCA based on node-to-node transitions into the model based on link-to-link transitions. In Section 3 we then incorporate the geometric rotations of paths into the link-based MCA model. Based on these preliminary analyses, in Section 4 we consider the model whose path set is restricted to simple paths only, and then the novel method to eliminate the cycle flows without computational burden is developed. In Section 5, we discuss the extension to the flow dependent model. Finally, we present our conclusions and discuss the future research.

1. Preliminaries - LOGIT Assignment through Markov Chain

Preliminary to the presentation of the new models, this section briefly reviews the Markov Chain Assignment that is consistent with LOGIT type route-choice model. For further detail, see **Akamatsu** (1996a,b,c), **Bell** (1995) and **Sasaki** (1965).

1.1. Networks

Our model is defined on a traffic network G [$\underline{N}$, $\underline{L}$] which has the set $\underline{N}$ of nodes, the set $\underline{L}$ of directed links and given set of origin-destination(OD) node pairs. The set $\underline{N}$ consists of three subset: the set of traversal nodes, N, that of origin nodes, O, and that of destination nodes, D. The number of nodes in each subset are n, g, and s, respectively. The link from node i to j is denoted as link (i, j). Each link in $\underline{L}$ has the travel cost, which is assumed to be fixed (i.e. *flow independent*). The extension to the flow dependent costs will be discussed in Section 5.

In the followings, each node in G is assumed to belong to only one of the subsets N, O, or D: $N \cap O = \varnothing, N \cap D = \varnothing$, and $O \cap D = \varnothing$. In addition, we assume that the number of links emanating from each origin is unique, and similarly, the number of links entering into each destination is unique. These assumptions are made only for the clarity of the presentation. In fact, they do not affect the generality of the subsequent models, since any traffic network can be modified so as to satisfies these assumptions without loss of generality (see Fig.1.1).

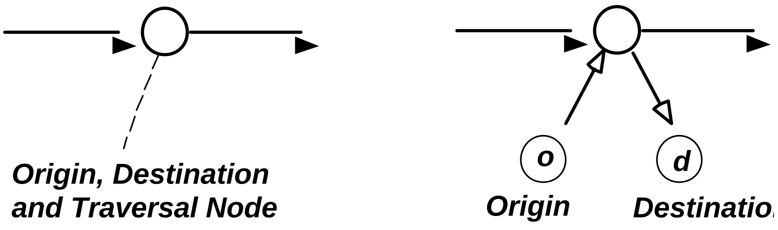

Fig. 1.1. (a) Original network. (b) Modified network.

1.2. Traffic Assignment through Markov Chain

We consider the network with multiple origins and single destination ("Many-to-One OD-pattern"). As can be seen in below, we can easily extend the model to the case of Many-to-Many OD-pattern by simply overlapping the flow pattern for each Many-to-One OD-pattern. The Markov Chain Assignment by **Sasaki**(1965) regards the nodes as the states in Markov chain, and the vehicles generated from origins are assumed to repeat the transition between the states according to the Markov chain rule. To formulate the model, let $\underline{Q}_{[d]}$ be the transition probabilities matrix with the following structure:

$$\underline{\mathbf{Q}}_{[d]} = \begin{array}{cc} \begin{array}{ccc} \mathit{1} & g & n \end{array} & \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \mathbf{Q}_1 \\ \mathbf{Q}_{2[d]} & 0 & \mathbf{Q}_{[d]} \end{bmatrix} & \begin{array}{c} \mathit{1} \\ g \\ n \end{array} \end{array} \tag{1.1}$$

where Q_1 is a $g \times n$ matrix whose (o, i) component is 1 if the origin o is connected with the traversal-node i, zero otherwise,

$Q_{2[d]}$ is an $n \times 1$ vector whose i th component is 1 if the destination d is connected with the traversal-node i, zero otherwise,

$Q_{[d]}$ is an $n \times n$ matrix whose (i, j) component denotes the (non-zero) transition probabilities, Q_{ij}, if traversal-node pair (i, j) is connected by a single link, zero otherwise.

Then, we easily see that the (i, j) component of $\underline{Q}_{[d]}{}^{k}$ is the sum of choice probabilities of k-walks paths (i.e. the paths consisting of k nodes) between node pair (i,j). Therefore, the node-choice probabilities (conditional on being generated from each node) are given by the matrix series $\mathrm{I} + \underline{Q}_{[dl} + \underline{Q}_{[dl}{}^{2} + \underline{Q}_{[dl}{}^{3} + \cdots$, and it converges to the following inverse matrix:

$$\mathrm{I} + \underline{Q}_{[d]} + \underline{Q}_{[d]}{}^{2} + \underline{Q}_{[d]}{}^{3} + \cdots = [\mathrm{I} - \underline{Q}_{[d]}]^{-1} = \begin{bmatrix} \mathbf{I} & 0 & 0 \\ \mathbf{P}_{3[d]} & \mathbf{I} & \mathbf{P}_{1[d]} \\ \mathbf{P}_{2[d]} & 0 & \mathbf{P}_{[d]} \end{bmatrix}, \tag{1.2a}$$

$$\mathrm{P}_{[d]} = [\mathrm{I} - \mathrm{Q}_{[d]}]^{-1}, \tag{1.2b}$$

$$\mathrm{P}_{1[d]} = \mathrm{Q}_1 \mathrm{P}_{[d]}, \quad \mathrm{P}_{2[d]} = \mathrm{P}_{[d]} \mathrm{Q}_{2[d]}, \quad \mathrm{P}_{3[d]} = \mathrm{Q}_1 \mathrm{P}_{[d]} \mathrm{Q}_{2[d]}, \tag{1.2c}$$

where $\mathrm{P}_{1[d]}$ is a $g \times n$ matrix whose (o, i) component, $P_{1[d]}$ ($i|o$), denotes the probability that a vehicle generated from origin o uses node i. Thus, we can obtain the link-choice probabilities by OD-pair, by substituting the $P_{1[d]}(i|o)$ obtained in (1.2) into the following definitional formula:

$$p_{ij}{}^{od} = P_{1[d]}(i|o) \bullet Q_{ij}, \tag{1.3}$$

where $p_{ij}{}^{od}$ denotes a probability that a vehicle with OD-pair (o,d) choose a link (i, j).

.

<u>1.3. Transition Probabilities Consistent with LOGIT type Route Choice</u>

At first sight, the MCA above has no background of behavior-theory, but it leads to the LOGIT based assignment: **Akamatsu** (1996a) showed that MCA yields the flow pattern that is consistent with the LOGIT type route choice model, if the transition probabilities are given by

$$Q_{ij} = Q_{[d]}(j|i) = W_{ij} V_{jd} / V_{id} \qquad or \qquad \mathrm{Q}_{[d]} = \mathrm{V}_{2[d]}{}^{-1} \mathrm{W} \, \mathrm{V}_{2[d]}, \tag{1.4}$$

where $\quad W_{ij} \equiv \exp[-q \, t_{ij}], \tag{1.5}$

$$V_{id} \equiv \sum_{r \in R_{\infty}^{id}} \exp[-q\, C_r^{id}], \tag{1.6}$$

t_{ij}: the cost of link (i, j), q : the sensitivity parameter of LOGIT model,

R_{∞}^{id} : the set of all the possible paths between traversal-node i to destination d,

$C_r^{kd} \equiv \sum_{ij \in L} t_{ij} d_{r,ij}^{kd}$: the cost of r th path from node k to destination d,

$\mathrm{V}_{2[d]}$ = an $n \times n$ diagonal matrix whose (i, i) diagonal component is V_{id}.

Note that we can deal with the Many-to-Many OD pattern by simply defining the transition probabilities for Many-to-One OD patterns by each origin.

How we calculate the V_{id} defined in (1.6) is not self-evident since the definition requires the summation with respect to infinite paths. Nevertheless, **Bell** (1995) showed that it can be obtained by the simple matrix operations as follows. Let $\underline{\mathrm{W}}$ be the "impedance" matrix with the following form:

$$\underline{\mathbf{W}} = \begin{array}{c} \begin{array}{ccc} s & g & n \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \mathbf{W}_1 \\ \mathbf{W}_2 & 0 & \mathbf{W} \end{bmatrix} \end{array} \begin{array}{c} \\ s \\ g \\ n \end{array} \tag{1.7}$$

where W_1 is a $g \times n$ matrix whose (o, i) component is 1 if the origin o is connected with the traversal-node i, zero otherwise,

W_2 is an $n \times s$ matrix whose (i, d) component is 1 if the destination d is connected with the traversal-node i, zero otherwise,

W is an $n \times n$ matrix whose (i, j) component is $W_{ij} = \exp[-q\, t_{ij}]$ if traversal-node pair (i, j) is connected by a link, zero otherwise.

Then, it follows that the (i, j) component of $\underline{\mathrm{W}}^{k}$ is the sum of $\exp[-q\, C_r^{ij}]$ over the k-walks paths between node pair (i, j). Therefore, the matrix **V** is given by $\mathrm{I} + \underline{\mathrm{W}} + \underline{\mathrm{W}}^2 + \underline{\mathrm{W}}^3 + \cdots$, and it converges to the following inverse matrix:

$$\mathrm{I} + \underline{\mathrm{W}} + \underline{\mathrm{W}}^2 + \underline{\mathrm{W}}^3 + \cdots = [\mathrm{I} - \underline{\mathrm{W}}]^{-1} = \begin{bmatrix} \mathbf{I} & 0 & 0 \\ \mathbf{V}_3 & \mathbf{I} & \mathbf{V}_1 \\ \mathbf{V}_2 & 0 & \mathbf{V} \end{bmatrix} \tag{1.8a}$$

$$\mathrm{V} = [\mathrm{I} - \mathrm{W}]^{-1}, \quad \mathrm{V}_1 = \mathrm{W}_1 \mathrm{V}, \quad \mathrm{V}_2 = \mathrm{V}\, \mathrm{W}_2, \quad \mathrm{V}_3 = \mathrm{W}_1 \mathrm{V}\, \mathrm{W}_2, \tag{1.8b}$$

where the (o, i) component of matrix $\mathrm{V}_{1,}$ the (j, d) component of matrix V_2 , and the (o, d) component of matrix V_3 are defined as $V_{oi} = \sum_r \exp[-q\, C_r^{oi}]$, $V_{jd} = \sum_r \exp[-q\, C_r^{jd}]$, and $V_{od} = \sum_r \exp[-q\, C_r^{od}]$, respectively.

1.4. LOGIT Assignment through Markov Chain

To sum up, we can obtain the link-choice probabilities that are consistent with LOGIT model by the following procedures:

Step 1: (a) Calculate the variable **V** by (1.8),
(b) Determine the transition matrix **Q** by substituting the **V** into (1.4).
Step 2: (a) Compute the node-choice probabilities by the Markov Chain formula (1.2),
(b) Calculate the link-choice probabilities by (1.3).

It is worth mentioning that the calculation in Step 2 reduces to the simple calculation as follows. Substituting (1.4) into (1.2), the node-choice probabilities yields

$$\begin{aligned} \mathrm{P}_{1[d]} &= \mathrm{Q}_1\,[\mathrm{I}-\mathrm{Q}_{[d]}]^{-1} = \mathrm{Q}_1\,[\mathrm{I}-\mathrm{V}_{2[d]}{}^{-1}\,\mathrm{W}\,\mathrm{V}_{2[d]}]^{-1} \\ &= \mathrm{Q}_1\,\mathrm{V}_{2[d]}{}^{-1}\,[\mathrm{I}-\mathrm{W}]^{-1}\,\mathrm{V}_{2[d]} \\ &= \mathrm{Q}_1\,\mathrm{V}_{2[d]}{}^{-1}\,\mathrm{V}\,\mathrm{V}_{2[d]}\,. \end{aligned} \tag{1.9a}$$

That is, the (o,i) component of $\mathrm{P}_{1[d]}$, $P_{1[d]}(i|o) = P(i|o,d)$, is

$$P(i|o,d) = \sum_k d_{ok}\,\{(1/V_{kd})\,V_{ki}\,V_{id}\} = V_{oi}\,V_{id}\,/\,V_{od}\,. \tag{1.9b}$$

Therefore, the the flow on link (i, j) with OD-pair od, x_{ij}^{od}, is given by the following formula without explicit calculation of the inverse matrix $[\mathrm{I}-\mathrm{Q}_{[d]}]^{-1}$:

$$\begin{aligned} x_{ij}^{od} &= q_{od}\,p_{ij}^{od} = q_{od}\,P(i|o,d)\bullet Q_{[d]}(j|\,i) = q_{od}\,(V_{oi}\,V_{id}\,/\,V_{od}\,)\bullet(W_{ij}\,V_{jd}\,/\,V_{id}) \\ &= q_{od}\,V_{oi}\,W_{ij}\,V_{jd}\,/\,V_{od}\,, \end{aligned} \tag{1.10}$$

where q_{od} is the OD-flow between o and d.

1.5. The Problem Caused by Cycle Flows

The model above has a solution (i.e. the left hand side of (1.8a) converges and all the components take positive value) if and only if the spectral radius of W, ρ(W), is less than unity:

$$\rho(\mathrm{W}) = \underset{i}{Max.}\,|l_i| < 1, \tag{1.11}$$

where l_i is the i th eigenvalue of matrix W. The condition (1.11), however, may not be satisfied in some cases. For example, when the network has many cycles with zero costs, the network can induce infinite cycle flows, which means the non-existence of the solution. There also may be possibilities that unrealistic cycle flows are assigned even if the model has a solution. With this problem in mind, we will develop methods for reducing / eliminating the cyclic flows in the subsequent sections.

2. Markov Chain Assignment based on Link-to-Link Transitions

2.1. Markov Chain Assignment on a Line-Graph

The cycle flows that have the worst effect on the assignment flow pattern in practical applications are "U-turn" flows in a pair of opposite-directional links that connect the same node-pair. For example, the network shown in Fig.2.1 can induce excessive U-turn flows that repeat going between links 4 and 6. This kind of cycle flows, however, can be easily eliminated by using the concept of "line-graph" (for the definition, see, for example, **WILSON**(1985), **TAKENAKA**(1989) etc.).

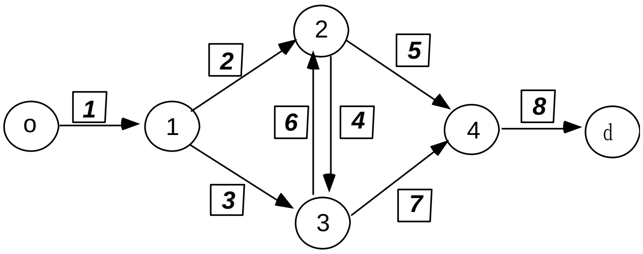

Fig. 2.1. An example network.

Let us consider a line-graph G*(N^*, L^*) of a given original network G($\underline{N}$, $\underline{L}$): each node in N^* has one-to-one correspondence to a link in $\underline{L}$, and the nodes in N^* are connected by links in L^* if the corresponding links in $\underline{L}$ are adjoining. Note here that the "U-turn" movement in G($\underline{N}$, $\underline{L}$) corresponds to a movement between a pair of nodes in N^*. Therefore, we construct L^* *eliminating the links that connect such node pairs in* N^*. Then any assignment models on the graph G*(N^*, L^*), clearly, can not generate the "U-turn" flows. Thus, by considering a Markov chain assignment on G*(N^*, L^*), we can obtain a link flow pattern explicitly avoiding the "U-turn" flows.

Fig.2.2 illustrates the line-graph G*(N^*, L^*) for the example network G($\underline{N}$, $\underline{L}$) shown in Fig.2.1. In the line-graph the nodes 4 and 6 are not mutually connected, and therefore, we see that the Markov chain assignment on G*(N^*, L^*) does not generate any cycle flows.

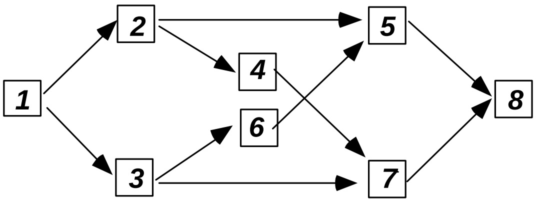

Fig. 2.2. A line-graph.

It is worth a mention in passing that this technique can be used not only for eliminating the cycles but also for representing turn-restrictions due to traffic control policies: we have only to eliminate the appropriate links in the line-graph corresponding to the turn-restrictions.

2.2. The Implementation of the Link based Markov Chain Assignment

In order to implement the idea above, we slightly modify the previous MCA model as follows. Note first that a Markov chain on G*(N^*, L^*) corresponds to the Markov chain based on "link-to-link" transitions on G($\underline{N}$, $\underline{L}$). Hence, it is natural to consider a "link-to-link" impedance matrix instead of the "node-to-node" matrix in the previous MCA model. Before giving the formal representation, we first classify the link set $\underline{L}$ into the following three subsets:

L^O : the set of links emanating from origins, where the number of elements is g,
L^D : the set of links entering into destinations, where the number of elements is s,
L : the set of the other links ($L^O \cup L^D \cup L = \underline{L}$), where the number of elements is ℓ.

We then define the link-to-link impedance matrix with the following structure:

$$\underline{\hat{\mathbf{W}}} = \begin{array}{c} \begin{array}{ccc} s & g & \ell \end{array} \\ \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & \hat{\mathbf{W}}_1 \\ \hat{\mathbf{W}}_2 & 0 & \hat{\mathbf{W}} \end{array}\right] \end{array} \begin{array}{c} \\ s \\ g \\ \ell \end{array} \qquad (2.1)$$

where the following matrix notation is used:

a) $\hat{\mathrm{W}}$ is an $\ell \times \ell$ matrix whose (a, b) component $\hat{W}_{ab}$ is exp(?q t_b) if the movement from link a to link b is permitted [when the movement involves the junction delay, d_{ab}, it is also appropriate to define $\hat{W}_{ab}$ = exp(?q (t_b +d_{ab}))], zero otherwise;

b) $\hat{\mathrm{W}}_1$ is a $g \times \ell$ matrix whose (o, a) component $\hat{W}_{oa}$ is exp(?q t_a) if link o in L^O is adjoining to link a in L , zero otherwise;

c) $\hat{\mathrm{W}}_2$ is an $\ell \times s$ matrix whose (a, d) component $\hat{W}_{ad}$ is 1 if link a in L is adjoining to link d in L^D, zero otherwise.

The definition of the impedance matrix is designed such that the (a,b) component of $\hat{\mathrm{W}}^k$ yields the sum of exp(?q C_r^{ab}) over the k+*1*-walks paths between link-pair (a,b), where C_r^{ab} denotes the cost of r th path between link a and link b in G($\underline{N}$, $\underline{L}$) excluding t_a. Accordingly, the Markov chain assignment based on the link-to-link transition probabilities defined by

$$Q(b\,|\,a) = \hat{W}_{ab}\frac{\hat{V}_{b\hat{d}}}{\hat{V}_{a\hat{d}}} \qquad \forall a,b \in L,\ \ \hat{d} \in L^D \qquad (2.2)$$

$$\hat{V}_{a\hat{d}} \equiv \sum_{r \in R_\infty^{a\hat{d}}} \exp[-q\, C_r^{a\hat{d}}] \qquad \hat{d} \in L^D \qquad (2.3)$$

yields the LOGIT type assignment. The proof almost parallels the case of the node-to-node MCA model, and it is omitted here. It also immediately follows that $\hat{\mathrm{V}} = [\hat{V}_{ab};\ \forall a,b \in L]$ can be obtained by

$$\hat{\mathrm{V}} = [\mathrm{I} - \hat{\mathrm{W}}]^{-1}, \qquad (2.4)$$

and that $\hat{\mathrm{V}}_1 \equiv [\hat{V}_{\hat{o}a}\,; \forall a \in L, \forall \hat{o} \in L^O]$, $\hat{\mathrm{V}}_2 \equiv [\hat{V}_{a\hat{d}}\,; \forall a \in L, \forall \hat{d} \in L^D]$, and $\hat{\mathrm{V}}_3 \equiv [\hat{V}_{\hat{o}\hat{d}}\,; \forall \hat{o} \in L^O$, $\forall \hat{d} \in L^D$] are, respectively, given by

$$\hat{\mathrm{V}}_1 = \hat{\mathrm{W}}_1 \hat{\mathrm{V}} \;\;,\;\; \hat{\mathrm{V}}_2 = \hat{\mathrm{V}}\,\hat{\mathrm{W}}_2 \quad \text{and} \quad \hat{\mathrm{V}}_3 = \hat{\mathrm{W}}_1 \hat{\mathrm{V}}\,\hat{\mathrm{W}}_2 \,. \tag{2.5}$$

Finally, the formula for the link flow can be obtained by utilizing the fact that the link-choice probability means the node-choice probability in the line-graph. That is, considering the correspondence to (1.9b) in the node-to-node MCA model, we have the following formula for the flow on link *a* with OD pair *od*:

$$x_a^{od} = q_{od}\; \hat{V}_{\hat{o}a} \hat{V}_{a\hat{d}} \,/\, \hat{V}_{\hat{o}\hat{d}} \,, \qquad \hat{o} \in L^O,\;\; \hat{d} \in L^D. \tag{2.6}$$

Thus, we can efficiently obtain the link flow pattern that is consistent with LOGIT model explicitly avoiding the “U-turn” flow.

It is expected in many practical applications that the modified MCA model can avoid the problem that the solution does not exist due to the excessive cycle flows. Nevertheless, the model may not necessarily yields the satisfactory flow pattern, since there still be a possibility that excessive flows are assigned on various possible cycles except “U-turn”. In the next section we will further devise another method for mitigating the cycle flows.

3. Markov Chain Assignment with Geometric Rotations

3.1. Rotation-angles between Links

Conventional stochastic assignment models assign the flows to many paths by some criteria based on the one-dimensional factor “travel time”. In contrast, various traffic surveys have reported that the share of “mazy” paths with many zigzaggings or turns are very small even if the travel times are moderate. In light of this fact, it is reasonable to introduce not only the travel time but also the geometric factors into the path-choice model.

To develop such a model, consider first the 2-dimentional Euclid geometric space to represent the geographic position of nodes in a traffic network. Then the links can be represented as vectors on 2-dimentional real number space R^2. In other word, we can regard a traffic network as a set of vectors on R^2. Using the vectors, we define the rotation-angle between mutually adjacent links *a* and *b* as follows:

$$w_{ab} = \mathrm{sgn}(\vec{a}, \vec{b}) \cdot \cos^{-1}\left(\frac{\vec{a} \cdot \vec{b}}{|\vec{a}|\,|\vec{b}|} \right), \qquad 0 \le w_{ab} \le p \,, \tag{3.1}$$

where $\vec{a}$ is a vector representing link *a* on R^2; and $\mathrm{sgn}(\vec{a}, \vec{b})$ = +1 if the movement from $\vec{a}$ to

$\vec{b}$ is right-handed turn, -1 otherwise (See Fig.3.1).

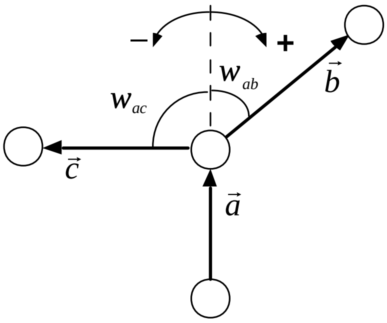

Fig. 3.1. The rotation-angle between mutually adjacent links

Then the total rotation-angle along r th path between OD pair od is naturally introduced as the sum of the absolute angles between adjoining links on the path:

$$D_r^{od} \equiv \sum_{(a,b)\in \Lambda} |w_{ab}|\, d_{r,ab}^{od}\ , \tag{3.2a}$$

where $d_{r,ab}^{od}$ is 1 if the mutually adjacent links a and b are on r th path between OD pair od, zero otherwise, Λ is the set of link-pairs where the two links are mutually adjacent. Similarly, the "net" total rotation-angle for the path is also defined as follows:

$$E_r^{od} \equiv \sum_{(a,b)\in \Lambda} w_{ab} d_{r,ab}^{od}\ . \tag{3.2b}$$

In addition, we define the rotation-angle between links that *are not mutually adjacent*. While the definition is basically the same as that for the mutually adjacent links, we should note the definition when two links are in a parallel position. The angle between the parallel links a and b is defined as follows:

Step 1: add two virtual links between the links a and b so as to construct a cycle (See Fig.3.2);

Step 2: if the cycle is right-handed then $w_{ab} = \pi$ and $w_{ba} = -\pi$,
otherwise $w_{ab} = -\pi$ and $w_{ba} = \pi$.

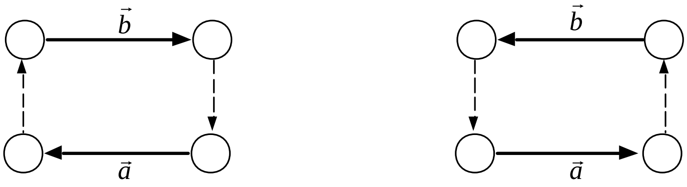

Fig.3.2. (a) Right-handed link-pair. (b) Left-handed link-pair.

3.2. LOGIT assignment with Geometric Rotations

To incorporate the geometric factors into the path-choice model, suppose that the systematic utility for the *r* th path between *o* and *d*, U_r^{od}, is represented as

$$U_r^{od} = -q\, C_r^{od} - s\, D_r^{od}, \tag{3.3}$$

where σ is a “sensitivity” parameter for the total rotation-angle. Then, by assuming the error term of the utility function to be i.i.d Gumbel distribution, the probability that users choose *r* th path between OD pair *od* yields

$$P_r^{od} = \frac{\exp[-q\, C_r^{od} - s\, D_r^{od}]}{\sum_{r \in R_\infty^{od}} \exp[-q\, C_r^{od} - s\, D_r^{od}]} = \frac{A_r^{od}}{\sum_{r \in R_\infty^{od}} A_r^{od}}, \tag{3.4a}$$

$$A_r^{od} \equiv \exp[-q\, C_r^{od} - s\, D_r^{od}], \tag{3.4b}$$

and the resulting link flow pattern is given by

$$x_a^{od} = q_{od} \sum_{r \in R_\infty^{od}} P_r^{od} d_{r,a}^{od}. \tag{3.5}$$

Since this model is basically a LOGIT model, we can construct the corresponding MCA. Unlike the previous MCA, we can expect that this MCA does not give rise to much cycle flows, since the paths with cycles have large negative utilities due to the rotation-angles.

The algorithm for this model is almost same with that for the previous MCA model. The only difference is the slight modification of the link impedance matrix: the (*a*,*b*) component of the matrix is defined as

$$\hat{W}_{ab} = \exp[-q\, t_b - s\, |w_{ab}|]. \tag{3.6}$$

The reason why this modification yields the LOGIT assignment model in (3.4a), (3.4b) and (3.5) can be easily understood from the fact that the (*a*,*b*) component of $\hat{\mathrm{W}}^k$ yields the sum of $\exp(-q\, C_r^{od} - s\, D_r^{od})$ over the *k+1*-walks paths between link-pair (*a*,*b*).

Thus, one may expect that the model combined with the technique in Section 2 greatly reduces cycle flows in many cases. The model, however, does not achieve our purpose, since it still remains a possibility of generating unrealistic flow patterns. For example, the model can not distinguish between a path with one cycle and a simple path with 4 rectangular turns, while almost all real users think the former path unreasonable. Recall here that the model shown above did not utilize the net total rotation-angles E_r^{od} defined in (3.2b) at all. Interestingly, the exploitation of the E_r^{od} gives us the innovative method to achieve the complete elimination of cycle flows, which will be presented in the next section.

4. Complete Elimination of Cycle Flows

4.1. LOGIT type Route Choice Model without Cycles

In this section we will consider the method for obtaining the link flow pattern according to the following LOGIT model:

$$P_r^{od} = \frac{\exp[-q\, C_r^{od} - s\, D_r^{od}]}{\sum_{r\in R^{od}(0)} \exp[-q\, C_r^{od} - s\, D_r^{od}]} = \frac{A_r^{od}}{\sum_{r\in R^{od}(0)} A_r^{od}}\,, \tag{4.1a}$$

$$x_a^{od} = q_{od} \sum_{r\in R^{od}(0)} P_r^{od} d_{r,a}^{od}\,, \tag{4.1b}$$

where the $R^{od}(0)$ is the set of simple paths between nodes o and d. Note that the path set in (4.1), unlike the MCA model in Section 3, consists of the simple paths only; the cycle flows are explicitly eliminated. The main focus here is, thus, to develop the method for achieving the complete elimination of cycle flows by exploiting the E_r^{ab} introduced in the previous section.

Just as in the case of the previous MCA, the method presented in this section consists of the following two steps:

Step 1: Calculate the matrix $\hat{\mathbf{V}}$ whose (a,b) component is defined as

$$\hat{V}_{ab} \equiv \sum_{r\in R^{ab}(0)} A_r^{ab}\,. \tag{4.2}$$

Step 2: Compute the link flows by

$$x_a^{od} = q_{od}\, X_a^{od} / \hat{V}_{od}\,, \tag{4.3a}$$

$$X_a^{od} \equiv \sum_{r\in R^{od}(0)} A_r^{od} d_{r,a}^{od}\,. \tag{4.3b}$$

Before going into the detailed discussions on the steps above, we will first examine the basic properties of the net rotation-angle E_r^{ab} in 4.2. Making use of the properties, we then develop a novel method for achieving Step 1. The considerations on the method is divided into two parts, which are presented in the subsequent sections 4.3 and 4.4, respectively. Finally, the method for calculating the X_a^{od} in Step 2 is presented in 4.5.

4.2. Basic Properties of the Path Rotation-angle

Let us consider the properties of the net rotation-angle defined in (3.2). First, it is easily observed that the rotation-angle of a path between links a and b, E_r^{ab}, is equal to $-w_{ba}$ if the path contains no cycles. For example, the path (a,b,c) in Fig.4.1 (a) contains no cycles, and the rotation-angle is $E^{ac} = w_{ab} + w_{bc} = -w_{ca}$ or $E^{ac} + w_{ca} = 0$. Next, consider the cases where the path contains cycles. In the example of Fig.4.1 (b), the path (a,b,c,d,e) contains one left-handed cycle, and the rotation-angle is $E^{ae} + w_{ea} = w_{ab} + w_{bc} + w_{cd} + w_{de} + w_{ea} = -2\pi$. Similarly, the path (a,b,c,d,b,c,d,e) has double left-handed cycles, and $E^{ae} + w_{ea} = -4\pi$. Thus, we see that $E_r^{ab} + w_{ba} = 2m\pi$ $(m=\pm 1,\pm 2,...)$ for the paths with cycles, and that the value of $E_r^{ab} + w_{ba}$ can be a "detector" that shows whether the path contains cycles or not. Note, however, that there is a particular case that the $E_r^{ab} + w_{ba}$ can be zero even if the path contains cycles: when we consider the path where the number of right-handed cycles, N_R , is equal to that of left-handed cycles, N_L , the $E_r^{ab} + w_{ba}$ yields zero since the rotation-angles for the respective cycles mutually cancel.

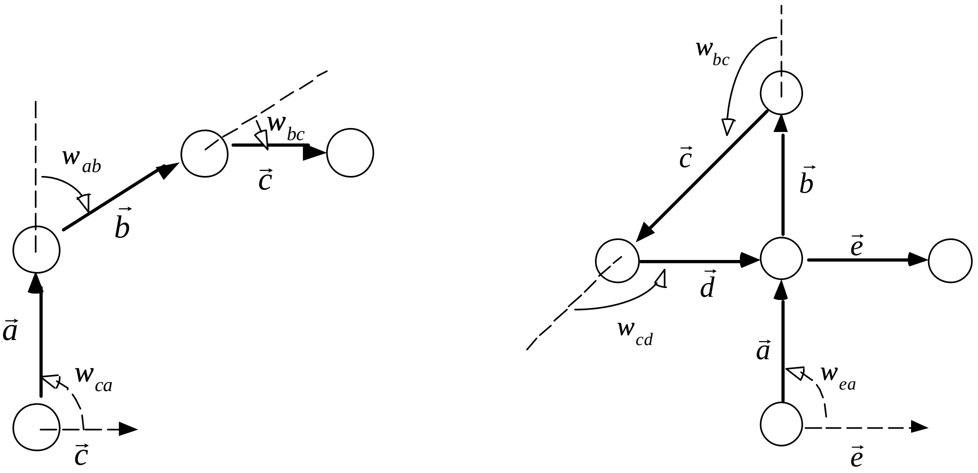

Fig. 4.1. (a) Path (a,b,c) (b) Path (a,b,c,d,e)

To give the formal representation of these properties, we denote by R_k^{ab} the set of k-walks paths between links a and b (paths from link a to link b consisting of k links). The path set can be classified into the following three subsets:

$R_k^{ab}(0)$: the subset of R_k^{ab}, which consists of the only paths containing no cycles,

$R_k^{ab}(1)$: the subset of R_k^{ab}, which consists of the paths such that $N_R \neq N_L$,

$R_k^{ab}(2)$: the subset of R_k^{ab}, which consists of the paths such that $N_R = N_L \neq 0$.

Similarly, we also classify the set of paths between links a and b, R^{ab}, into three subsets:

$$R^{ab}(m) \equiv \bigcup_{k=1}^{\infty} R_k^{ab}(m) \quad \text{for } m=0,1, \text{ and } 2.$$

Then, the path rotation-angle satisfies the following:

$$E_r^{ab} + w_{ba} = \begin{cases} 0 & r \in R^{ab}(0) & \text{(4.4a)} \\ 2mp \quad (m = \pm 1, \pm 2, \dots) & r \in R^{ab}(1) & \text{(4.4b)} \\ 0 & r \in R^{ab}(2) & \text{(4.4c)} \end{cases}$$

In the subsequent secction 4.3, we first consider the method for calculating the sum of A_r^{ab} for the path set $R^{ab}(0) \cup R^{ab}(2)$ (i.e. we eliminate the sum for $R^{ab}(1)$); and then we proceed to the method to further exclude the path set $R^{ab}(2)$ as well as $R^{ab}(1)$ in section 4.4.

4.3. Elimination of the Path Set R(1)

Suppose for the moment that the path set $R^{od}(2)$ is excluded from R^{od} by an appropriate method: we assume that R^{od} = $R^{od}(0) \cup R^{od}(1)$. Then, the $\hat{V}_{od}$ defined in (4.2) can be re-written as follows:

$$\hat{V}_{od} = \sum_{r \in R^{od}(0)} A_r^{od} = \sum_{r \in R^{od}} d(E_r^{od} + w_{do}) \cdot A_r^{od} \ , \qquad (4.5a)$$

where $d(z) \equiv \begin{cases} 1 & if \quad z = 0 \\ 0 & otherwise \quad (i.e. \quad z = 2mp\ , \ \ m = \pm 1, \pm 2, ...) \end{cases}$. (4.5b)

Recall here the Fourier series expansion of Dirac's delta-function:

$$d(z) = \frac{1}{T} + \frac{2}{T} \sum_{n=1}^{\infty} \cos\left(\frac{2np}{T} z \right)$$

$$= \frac{1}{T} \sum_{n=-\infty}^{\infty} \exp\left(i \frac{2np}{T} z \right) \quad \text{for} \ \ -T/2 \le z \le T/2 \ , \qquad (4.6)$$

where *i* denotes an imaginary unit. From (4.5) and (4.6), we see that $\hat{V}_{od}$ is represented by the (complex) exponential function, which can be decomposed into link variables. Thus, if we construct the MCA model based on the appropriate impedance matrix that is consistent with (4.6), the sum of A_r^{ab} for $R^{ab}(1)$ would naturally be eliminated. The procedure presented below is, in essence, the realization of this idea.

Let $\hat{\mathbf{W}}(n, N)$ be the "*complex* impedance" matrix with the following (*a*, *b*) component:

$$\hat{W}_{ab}(n, N) = \exp[-q\, t_b - s\, |w_{ab}| + i w_{ab} \frac{n}{N}] \ = W_{ab}\, Y_{ab}(n,N) \qquad (4.7)$$

where $W_{ab} \equiv \exp[-q\, t_b - s\, |w_{ab}|]$, $Y_{ab}(n, N) \equiv \exp[i w_{ab} \frac{n}{N}]$, and (*n*,*N*) are the integers satisfying $1 \le n \le N$. Then, the (*a*, *b*) component of the *k* th power of the matrix, $\hat{W}_{ab}(n,N)^k$, yields

$$\hat{W}_{ab}(n, N)^k = \sum_{r \in R_{k+1}^{ab}} A_r^{ab} \exp[i\, E_r^{ab} \frac{n}{N}] \ , \qquad (4.8)$$

where R_{k+1}^{ab} is the set of *k+1-walks* paths between links *a* and *b*. Therefore, it immediately follows from (4.4a) and (4.4c) that

$$\hat{W}_{ab}(n, N)^k \ Y_{ba}(n, N) = \sum_{r \in R_{k+1}^{ab}} A_r^{ab} \exp[i\, (E_r^{ab} + w_{ba}) \frac{n}{N}]$$

$$= \sum_{r \in R_{k+1}^{ab}(0)} A_r^{ab} + \sum_{r \in R_{k+1}^{ab}(2)} A_r^{ab} + \sum_{r \in R_{k+1}^{ab}(1)} A_r^{ab} \exp[i\, (E_r^{ab} + w_{ba}) \frac{n}{N}] \ . \qquad (4.9)$$

The present subject here is to eliminate the third term of the r.h.s. of (4.9), which leads us to exclude the summation with respect to the path set $R^{ab}(1)$ in the calculation of $\hat{\mathbf{V}}$. In order to construct the elimination method, recall the fact that the value of $E_r^{ab} + w_{ba}$ in the third term must be $2mp$, ($m = \pm 1,2,...$). Considering this property and the (4.6), we have

$$\frac{1}{N}\sum_{n=1}^{N}\sum_{r\in R_{k+1}^{ab}(1)} A_r^{ab}\exp[i\,(E_r^{ab}+w_{ba})\frac{n}{N}] = \sum_{r\in R_{k+1}^{ab}(N)} A_r^{ab}\,, \tag{4.10}$$

where $R_{k+1}^{ab}(N)$ is the subset of $R_{k+1}^{ab}(1)$ such that $E_r^{ab} + w_{ba} = (2Np\,)\times \ell$, ($\ell = \pm 1, \pm 2,...$).

Table 4.1 shows the simple numerical example for (4.10) when there is only a single route, $N = 3$, and A_r^{ab}=1. The number in each cell denotes the value of

$$Re(\exp[\,i\,z\frac{n}{N}]) = COS\,(\,z\frac{n}{N}\,) = COS\,(2\,mp\frac{n}{N}),$$

where $z\alpha E_r^{ab} + w_{ba}$, and the rows and the columns correspond to n = 1,2,3 and m = 1,2,..., respectively. The final row denotes the $\Sigma_n\, COS\,(z\,n/N)$) for $z = 2mp$ (m =0,1,2,...). As seen in the table, the $\Sigma_n\, COS\,(z\,n/N)$) yields zero, except the cases where $z = (2Np\,)\times \ell = 0$, $6p$, $12p$,.... Thus, the r.h.s. of the (4.10) reduces to the sum over the paths such that $E_r^{ab} + w_{ba} = (2Np\,)\times \ell$.

Table 4.1. The numerical example of (4.10):
$COS\,(z\,n/N)$ and $\Sigma_n\, COS\,(z\,n/N)$ for $z\alpha E_r^{ab} + w_{ba} = 2\,mp$ (m =0,1,2,...).

n ↓	m → $z\alpha E_r^{ab} + w_{ba}$ →	0 0	1 2π	2 4π	3 6π	4 8π	5 10π	6 12π	...
1	$COS\,(\,z/3)$ →	1.0	-0.5	-0.5	1.0	-0.5	-0.5	1.0	...
2	$COS\,(2z/3)$ →	1.0	-0.5	-0.5	1.0	-0.5	-0.5	1.0	...
3	$COS\,(3z/3)$ →	1.0	1.0	1.0	1.0	1.0	1.0	1.0	...
Sum.		3.0	0.000	0.000	3.0	0.000	0.000	3.0	...

Note that the period (of z) that $\Sigma_n\, COS\,(z\,n/N)$) takes non-zero value grows as we increase N. Hence, by letting N be infinity in (4.10), we can eliminate the summation for the $R_{k+1}^{ab}(1)$:

$$\lim_{N\to\infty}\frac{1}{N}\left(\sum_{n=1}^{N}\sum_{r\in R_{k+1}^{ab}(1)} A_r^{ab}\exp[i\,(E_r^{ab}+w_{ba})\frac{n}{N}]\right) = 0\,. \tag{4.11}$$

Combining (4.9) and (4.11), we have

$$\lim_{N\to\infty}\frac{1}{N}\left(\sum_{n=1}^{N} W_{ab}(n,N)^k\;Y_{ba}(n,N)\right) = \sum_{r\in R_{k+1}^{ab}(0)} A_r^{ab} + \sum_{r\in R_{k+1}^{ab}(2)} A_r^{ab}\,, \tag{4.12}$$

and furthermore, summing up (4.12) for $k=1,2,...$, gives us

$$\lim_{N\to\infty}\frac{1}{N}\left(\sum_{k=1}^{\infty}\sum_{n=1}^{N}W_{ab}(n,N)^k\ \ Y_{ba}(n,N)\right)=\sum_{r\in R^{ab}(0)}A_r^{ab}+\sum_{r\in R^{ab}(2)}A_r^{ab}\ .$$

Namely, the sum of A_r^{ab} for the path set $R^{ab}(0)\cup R^{ab}(2)$ (i.e. the path set where the $R^{ab}(1)$ is excluded) is given by the (a, b) component of the following matrix:

$$\lim_{N\to\infty}\frac{1}{N}\sum_{n=1}^{N}\Big(\mathrm{I}*\mathrm{Y}(n,N)^T+\hat{\mathrm{W}}(n,N)*\mathrm{Y}(n,N)^T+\hat{\mathrm{W}}(n,N)^2*\mathrm{Y}(n,N)^T+\cdots\Big)$$

$$=\lim_{N\to\infty}\frac{1}{N}\sum_{n=1}^{N}\Big([\mathrm{I}-\hat{\mathrm{W}}(n,N)]^{-1}*\mathrm{Y}(n,N)^T\Big),\tag{4.13}$$

where $\mathrm{X}*\mathrm{Y}\equiv$ the component-wise multiplication of matrices X=[X_{ab}] and Y=[Y_{ab}] (=the matrix whose (a,b) component is $X_{ab}Y_{ab}$), Y(n,N)$\equiv$the matrix whose (a,b) component is $Y_{ab}(n,N)$.

4.4. Elimination of the Path Set R(2)

Let us extend the previous method so as to eliminate the path set $R^{ab}(2)$ as well as $R^{ab}(1)$. *Suppose now that the set of k-walks paths containing no cycles,*$\{R_k^{ab}(0)\ \ \forall(a,b)\}$, *is given.* Then we can construct the $k+1$-walks paths by adding only one link to the paths in $\{R_k^{ab}(0)\ \ \forall(a,b)\}$. We denote the set of the $k+1$-walks paths by $\{\tilde{R}_{k+1}^{ab}(0)\ \ \forall(a,b)\}$. Clearly, the number of cycles contained in each $k+1$- walks path in $\{\tilde{R}_{k+1}^{ab}(0)\ \ \forall(a,b)\}$ is at most one. Hence, the following holds for the paths in $\{\tilde{R}_{k+1}^{ab}(0)\ \ \forall(a,b)\}$:

$$E_r^{ab}+w_{ba}=0 \qquad \textit{if the path contains no cycles,}\tag{4.14a}$$

$$E_r^{ab}+w_{ba}=\pm 2p \qquad \textit{otherwise.}\tag{4.14b}$$

Consequently, *if the path set in the summation in (4.8) was not* R_{k+1}^{ab} *but* $\tilde{R}_{k+1}^{ab}(0)$, (4.12) would have reduced to the summation with respect to the only path set $R_{k+1}^{ab}(0)$ (i.e. the terms for $R_{k+1}^{ab}(2)$ as well as $R_{k+1}^{ab}(1)$ vanishes). Besides, since $\left|E_r^{ab}+w_{ba}\right|\le 2p$ for the path set $\tilde{R}_{k+1}^{ab}(0)$, it is not required to let N in (4.12) be infinity: *setting N=2 is enough for eliminating the cycles.*

The idea above can be stated more formally as follows. *Given the k-walks path set* $\{R_k^{ab}(0)\ \ \forall(a,b)\}$, define a matrix $\hat{\mathrm{W}}(n,N)^{[k-1]}$ whose (a,b) component is

$$\hat{W}_{ab}(n,N)^{[k-1]}\equiv\sum_{r\in R_k^{ab}(0)}A_r^{ab}\exp[iE_r^{ab}\frac{n}{N}]=\sum_{r\in R_k^{ab}(0)}\hat{A}_r^{ab}(n,N),\tag{4.15}$$

where $\hat{A}_r^{ab}(n,N)\equiv A_r^{ab}\exp[i\,E_r^{ab}\,\frac{n}{N}]$. Then, the ($a$, b) component of $\hat{\mathrm{W}}(n{,}2)^{[k-1]}\hat{\mathrm{W}}(n{,}2)$ yields

$$\sum_{p\in A(a)}\hat{W}_{ap}(n{,}2)^{[k-1]}\hat{W}_{pb}(n{,}2)=\sum_{r\in\tilde{R}_{k+1}^{ab}(0)}\hat{A}_r^{ab}(n{,}2),\tag{4.16}$$

where $A(a)$ denotes the set of links emanating from the end-node of link a (Note that the $\hat{\mathrm{W}}(n,2)^{[k-1]}\hat{\mathrm{W}}(n,2)$ is *not* equal to the $\hat{\mathrm{W}}(n,2)^{k}$ *nor* the $\hat{\mathrm{W}}(n,2)^{[k]}$ while it plays a similar roll with the $\hat{\mathrm{W}}(n,2)^{k}$ in (4.12)). Applying the same argument as (4.12) to (4.16), we see that the sum of A_r^{ab} for the path set $R_{k+1}^{ab}(0)$ is given by the (a,b) component of the following matrix:

$$\tilde{\mathrm{W}}^{[k]} \equiv \frac{1}{2}\sum_{n=1}^{2}\left(\hat{\mathrm{W}}(n,2)^{[k-1]}\hat{\mathrm{W}}(n,2) * \mathrm{Y}(n,2)^{T}\right). \tag{4.17}$$

Since the $\hat{V}_{ab}$ in (4.2) is defined as the sum of A_r^{ab} for the path set $R^{ab}(0) = \overset{\infty}{\underset{k=1}{\cup}} R_k^{ab}(0)$, the matrix $\hat{\mathrm{V}}$ with the component $\hat{V}_{ab}$ can be obtained by simply summing up (4.17) with respect to k. In addition, considering the fact that the number of links contained in the path without no cycles is at most ℓ (=the number of links in L), we have

$$\hat{\mathrm{V}} = \mathrm{I} + \tilde{\mathrm{W}}^{[1]} + \tilde{\mathrm{W}}^{[2]} + \cdots + \tilde{\mathrm{W}}^{[\ell-1]} = \sum_{k=0}^{\ell-1} \tilde{\mathrm{W}}^{[k]}, \tag{4.18}$$

where $\tilde{\mathrm{W}}^{[0]} \equiv \mathrm{I}$ (a unit matrix).

We showed thus far the method for obtaining $\hat{\mathrm{V}}$ when the path set $\{R_k^{ab}(0)\ \forall(a,b)\}$ or the matrix $\hat{\mathrm{W}}(n,2)^{[k-1]}$ are given in calculating $\tilde{\mathrm{W}}(n,2)^{[k]}$. In the following, we consider the method for obtaining the matrix $\hat{\mathrm{W}}(n,2)^{[k-1]}$ such that *it works successively in ascending order of k* : first, we set $\hat{\mathrm{W}}(n,2)^{[1]} := \hat{\mathrm{W}}(n,2)$ for *k=1*; and then, for each *k*≧*2*, we calculate the matrix $\hat{\mathrm{W}}(n,2)^{[k]}$ using the $\hat{\mathrm{W}}(n,2)^{[k-1]}$ previously obtained. Thus, the present subject is to constitute the appropriate formula for the successive calculation for *k*≧*2.*

To find the formula, let us first compare the definition of $\hat{\mathrm{W}}(n,2)^{[k]}$ and the $\hat{\mathrm{W}}(n,2)^{[k-1]}\hat{\mathrm{W}}(n,2)$: the (a,b) component of $\hat{\mathrm{W}}(n,2)^{[k-1]}\hat{\mathrm{W}}(n,2)$ is given by (4.16), and that of $\hat{\mathrm{W}}(n,2)^{[k]}$ is defined as

$$\hat{W}_{ab}(n,2)^{[k]} \equiv \sum_{r \in R_{k+1}^{ab}(0)} \hat{A}_r^{ab}(n,2). \tag{4.15'}$$

Note here that the set $\tilde{R}_{k+1}^{ab}(0)$ in (4.16) can be classified into two subsets $R_{k+1}^{ab}(0)$ and $\tilde{R}_{k+1}^{ab}(1)$:

$$\tilde{R}_{k+1}^{ab}(0) = \tilde{R}_{k+1}^{ab}(1) \cup R_{k+1}^{ab}(0) \quad \text{and} \quad \tilde{R}_{k+1}^{ab}(1) \cap R_{k+1}^{ab}(0) = \varnothing,$$

where the path set $\tilde{R}_{k+1}^{ab}(1)$ is defined as $\tilde{R}_{k+1}^{ab}(0) \cap R_{k+1}^{ab}(1)$ (see Fig.4.2). Thus it follows that

$$\begin{aligned}\sum_{p \in A(a)} \hat{W}_{ap}(n,2)^{[k-1]}\hat{W}_{pb}(n,2) - \hat{W}_{ab}(n,2)^{[k]} &= \sum_{r \in \tilde{R}_{k+1}^{ab}(0)} \hat{A}_r^{ab}(n,2) - \sum_{r \in R_{k+1}^{ab}(0)} \hat{A}_r^{ab}(n,2) \\ &= \sum_{r \in \tilde{R}_{k+1}^{ab}(1)} \hat{A}_r^{ab}(n,2) \end{aligned} \tag{4.19}$$

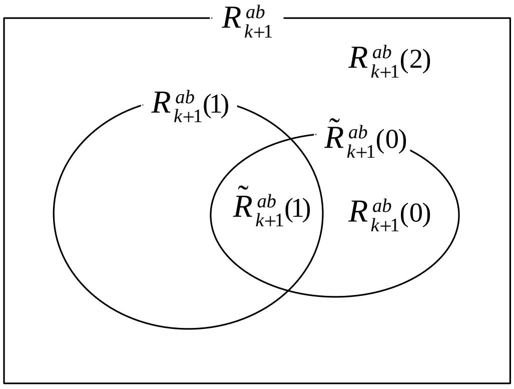

Fig. 4.2. The classification of *k+1*-walks paths.

Therefore, given a matrix $\mathrm{Z}(n,2)^{[k]}$ whose (a,b) component is defined as

$$\mathrm{Z}_{ab}(n,2)^{[k]} \equiv \sum_{r\in\tilde{R}^{ab}_{k+1}(1)} \hat{A}^{ab}_{r}(n,2)\,, \tag{4.20}$$

we have the following successive equation for calculating $\hat{\mathrm{W}}(n,2)^{[k]}$ in ascending order of k:

$$\hat{\mathrm{W}}(n,2)^{[k]} := \hat{\mathrm{W}}(n,2)^{[k-1]}\hat{\mathrm{W}}(n,2) - \mathrm{Z}(n,2)^{[k]}\,. \tag{4.21}$$

The $\mathrm{Z}_{ab}(n,2)^{[k]}$ can be calculated without explicitly dealing with path variables, while the definition in (4.20) is represented by the summation with respect to the path set $\tilde{R}^{ab}_{k+1}(1)$. Recall now the particular property for $\tilde{R}^{ab}_{k+1}(1)$: if the *k+1*-walks path $a \to \cdots \to p \to b$ belongs to $\tilde{R}^{ab}_{k+1}(1)$,

$$\sum_{n=1}^{2} \hat{W}_{ap}(n,2)^{[k-1]}\hat{W}_{pb}(n,2)\,Y_{ba}(n,2) = 0\,. \tag{4.22}$$

For the concrete example, consider two *5*-walks paths between links a and b as shown in Fig.4.3. The path $a \to \cdots \to p \to b$ belongs to $\tilde{R}^{ab}_{5}(1)$, where we can examine that (4.22) holds. On the other hand, for another path ($a \to \cdots \to q \to b$) that belong to $R^{ab}_{5}(0)$, it is easily seen that (4.22) does not hold.

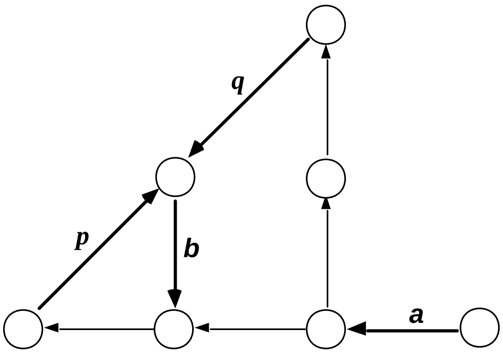

Fig. 4.3. Example network to consider (4.22).

From this property, the $Z_{ab}(n,2)^{[k]}$ is, clearly, computed by

$$Z_{ab}(n,2)^{[k]} := \sum_{p \in \Psi_{ab}} \hat{W}_{ap}(n,2)^{[k-1]} \hat{W}_{pb}(n,2), \qquad (4.23)$$

where Ψ_{ab} is the set of links $\{p\}$ such that (4.22) hold. Thus, the $\hat{W}(n,2)^{[k]}$ can be successively obtained in ascending order of k by the formula (4.21)-(4.23).

4.5. Calculation of Link Flows

We are now in a position to discuss the method for obtaining the link flows. For the calculation we can *not* use the formula for the MCA model presented in Section 2:

$$x_a^{od} = q_{od}\, \hat{V}_{oa}\hat{V}_{ad} / \hat{V}_{od}, \qquad o \in L^O,\ d \in L^D. \qquad (2.6)$$

The reason is that the denominator of (2.6) contains the term for the paths with cycles. More specifically, the multiplication of $\hat{V}_{oa}$ and $\hat{V}_{ad}$ yields

$$\hat{V}_{oa}\hat{V}_{ad} = \left(\sum_{r \in R^{oa}(0)} A_r^{oa} \right)\left(\sum_{r \in R^{ad}(0)} A_r^{ad} \right) = \sum_{r \in R_a^{od}} A_r^{od}, \qquad (4.24)$$

where R_a^{od} is the set of paths in which each path is constructed by connecting the following two kinds of paths: the paths between links o and a in $R^{oa}(0)$, and the paths between links a and d in $R^{ad}(0)$. Note that the path ($o \to \cdots \to a \to \cdots \to d$) in R_a^{od} can contain cycles, even if both the paths ($o \to \cdots \to a$) and ($a \to \cdots \to d$) contain no cycles (for the simple example, see Fig. 4.4). Therefore, we must replace the $\hat{V}_{oa}\hat{V}_{ad}$ in (2.6) with the variable X_a^{od} defined as

$$X_a^{od} \equiv \sum_{r \in R^{od}(0)} A_r^{od} d_{r,a}^{od} = \sum_{r \in R_a^{od}(0)} A_r^{od}, \qquad (4.25)$$

where $R_a^{od}(0)$ denotes the subset of R_a^{od} whose elements are restricted to the simple paths.

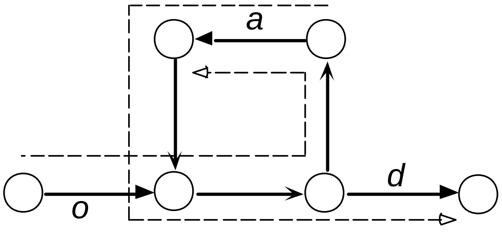

Fig. 4.4. The path ($o \to \cdots \to a \to \cdots \to d$) containing a cycle.

To consider the method for calculating the X_a^{od}, we first classify the set $R_a^{od}(0)$ by the number of links contained in the paths; we denote by $R_a^{od}(0{:}\, j+k)$ the set of the simple paths between o and d where each path is constructed by connecting the following two kinds of paths:

the j-walks paths between o and a, and the k-walks paths between a and d. Corresponding to this classification, we define $X_a^{od\,[j+k]}$ as the sum of A_r^{od} for the paths in $R_a^{od}(0{:}\,j{+}k)$:

$$X_a^{od\,[j+k]} \equiv \sum_{r\in R_a^{od}(0:j+k)} A_r^{od} \,. \tag{4.26}$$

Clearly,

$$X_a^{od} = \sum_{j=1}^{\ell}\sum_{k=1}^{\ell-j} X_a^{od\,[j+k]} \,. \tag{4.27}$$

In precisely the same fashion, we also classify the path set R_a^{od}, and we denote by $R_a^{od}(j{+}k)$ the subset corresponding to the subset $R_a^{od}(0{:}\,j{+}k)$ for the set $R_a^{od}(0)$.

The key to calculate the $X_a^{od\,[j+k]}$ is the fact that the number of cycles contained in any paths in $R_a^{od}(j{+}k)$ is at most one. From this fact, we see that the sum of A_r^{od} for the paths in $R_a^{od}(0{:}\,j{+}k)$, $X_a^{od\,[j+k]}$, can be obtained by subtracting the terms such that $E_r^{ab} + w_{ba} = \pm 2p$ from the sum for the paths in $R_a^{od}(j{+}k)$. Hence, by applying the similar method as shown in 4.4, we obtain

$$X_a^{od\,[j+k]} = \frac{1}{2}\sum_{n=1}^{2}\left(\hat{W}_{oa}(n,2)^{[j]}\hat{W}_{ad}(n,2)^{[k]}Y_{do}(n,2)\right). \tag{4.28}$$

Substituting this into (4.27), we have

$$\begin{aligned} X_a^{od} &= \frac{1}{2}\sum_{j=1}^{\ell}\sum_{k=1}^{\ell-j}\left\{\sum_{n=1}^{2}\left(\hat{W}_{oa}(n,2)^{[j]}\hat{W}_{ad}(n,2)^{[k]}Y_{do}(n,2)\right)\right\} \\ &= \frac{1}{2}\sum_{n=1}^{2} Y_{do}(n,2)\sum_{j=1}^{\ell}\sum_{k=1}^{\ell}\hat{W}_{oa}(n,2)^{[j]}\hat{W}_{ad}(n,2)^{[k]} \\ &= \frac{1}{2}\sum_{n=1}^{2} Y_{do}(n,2)\ \tilde{V}_{oa}(n,2)\ \tilde{V}_{ad}(n,2)\,, \end{aligned} \tag{4.29}$$

where

$$\tilde{V}_{ab}(n,2) \equiv \sum_{k=1}^{\ell}\hat{W}_{ab}(n,2)^{[k]} \,. \tag{4.30}$$

As shown in Section 4.4, the $\hat{W}_{ab}(n,2)^{[k]}$ can be computed in ascending order of k, and therefore, the $\tilde{V}_{ab}(n,2)$ in (4.29) also can be easily obtained. Thus, we see that the link flows can be calculated by the following formula:

$$x_a^{od} = \frac{1}{2} q_{od}\sum_{n=1}^{2}\tilde{V}_{oa}(n,2)\,\tilde{V}_{ad}(n,2)\,Y_{do}(n,2)\,/\,\hat{V}_{od}\,, \quad o\in L^O,\ d\in L^D. \tag{4.31}$$

5. Extension to the Stochastic Equilibrium Model

In this section we first discuss the implementation issues of the model presented in the previous section, and then we extend the model and algorithm to the flow dependent case.

5.1. The Outline of the Assignment Algorithm

The considerations for the assignment over a path set $R(0)$ so far can be summarized as the following algorithm:

Step 0: Initialization

```
for each $(a,b)$ in $\underline{L} \times \underline{L}$ do begin
    $\hat{V}_{ab} := 0$;   $\tilde{W}_{ab}^{[1]} := 0$;
    for $n = 1$ to $2$ do begin
        $Y_{ab}(n) := \exp[i\ w_{ab} / n]$;   $\hat{W}_{ab}(n)^{[1]} := \exp[-q\ t_b - s\,|w_{ab}|]\,Y_{ab}(n)$;      (5.1)
    end
end;
```

Step 1: Calculation of Matrix $[\hat{V}_{ab}]$

```
for $k = 2$ to $\ell$ do begin
    for each $(a,b)$ in $\underline{L} \times \underline{L}$ do begin
        $\hat{W}_{ab}(1)^{[k]} := 0$;  $\hat{W}_{ab}(2)^{[k]} := 0$;  $\tilde{W}_{ab}^{[k]} := 0$;
        for each $p$ in $\mathrm{A}(a)$ do begin
            $X := 0$;
            for $n = 1$ to $2$ do begin
                $w_p(n) := \hat{W}_{ap}(n)^{[k-1]}\,\hat{W}_{pb}(n)^{[1]}$ ;   $X := X + w_p(n)\ Y_{ba}(n)$      (5.2)
            end;
            for $n = 1$ to $2$ do   if $X \neq 0$ then $\hat{W}_{ab}(n)^{[k]} := \hat{W}_{ab}(n)^{[k]} + w_p(n)$      (5.3)
        end;
        for each $q$ in $\mathrm{A}(a)$ do   for $n = 1$ to $2$ do $\tilde{W}_{ab}^{[k]} := \tilde{W}_{ab}^{[k]} + w_q(n) Y_{bq}(n)$ ;      (5.4)
        $\hat{V}_{ab} := \hat{V}_{ab} + \tilde{W}_{ab}^{[k]} / 2$;      (5.5)
        for $n = 1$ to $2$ do $\tilde{V}_{ab}(n) := \tilde{V}_{ab}(n) + \hat{W}_{ab}(n)^{[k]}$      (5.6)
    end
end;
```

Step 2: Calculation of Link Flows by OD-pairs

```
for each $(o,d)$ in $L^O \times L^D$ do
    for each $a$ in $\underline{L}$ do $x_a^{od} := \frac{1}{2} q_{od} \sum_{n=1}^{2} \tilde{V}_{oa}(n)\ \tilde{V}_{ad}(n)\ Y_{do}(n) \big/ \hat{V}_{od}$ ;      (5.7)
```

5.2. Some Improvement of the Algorithm

In 5.1, we showed only a prototype of the algorithm. For the practical applications, however, the algorithm should be improved from the view point of both the required storage and the computational effort. First, we do not have to store $\hat{W}(n)^{[k]}_{ab}$ or $\tilde{W}^{[k]}_{ab}$ for all k : obviously, only the temporal $\tilde{W}^{[k]}_{ab}$ should be stored. Secondly, it is expected in many cases that we can truncate the algorithm before k reaches ℓ, since the maximum number of links contained in any simple paths in practical networks is usually much smaller than ℓ. Finally, it is not necessarily required for the calculation of $\{x_{ij}^{od}\}$ to store $\{\hat{V}_{ab}\ \forall(a,b) \in \underline{L} \times \underline{L}\}$; storing $\{\hat{V}_{oa}\ \forall a \in L, \forall o \in L^O\}$ and $\{\hat{V}_{ad}\ \forall a \in L, \forall d \in L^D\}$ is advantageous from the view point of the required storage. From these considerations, the steps 1 and 2 in 5.1 can be modified as follows:

Step 1 (a): Calculation of Matrix $[\hat{V}_{oa}]$ for each origin *o*

```
for each $o$ in $L^O$ do begin   $flag$ := 1;
   while $flag$ = 1 do begin   $flag$ := 0;
      for each $a$ in $L$ do begin   $W_a(1)$ := 0;  $W_a(2)$ := 0;  $Z$ :=0;
         for each $p$ in A($a$) do begin   $X$ := 0;
            for   $n$ = 1 to 2 do begin
               $w_p(n) := W_p(n)\hat{W}_{pa}(n)^{[1]}$ ; $X := X + w_p(n)\, Y_{pa}(n)$                  (5.2a)
            end;
            for   $n$ = 1 to 2 do   if $X \neq 0$   then   $W_a(n) := W_a(n) + w_p(n)$            (5.3a)
         end;
         for each $q$ in A($a$) do      for $n$ = 1 to 2 do    $Z := Z + w_q(n)Y_{qo}(n)$ ;         (5.4a)
         $\hat{V}_{oa} := \hat{V}_{oa} + Z/2$ ;      if   $Z \neq 0$ then   $flag$ := 1;
         for $n$ = 1 to 2 do  $\tilde{V}_{oa}(n) := \tilde{V}_{oa}(n) + W_a(n)$                       (5.6a)
      end
   end
end;
```

Step 1 (b): Calculation of Matrix $[\hat{V}_{ad}]$ for each destination *d*

```
for each $d$ in $L^D$ do begin   $flag$ := 1;
   while $flag$ = 1 do begin   $flag$ := 0;
      for each $a$ in $L$ do begin   $W_a(1)$ := 0;  $W_a(2)$ := 0;  $Z$ :=0;
         for each $p$ in A($a$) do begin   $X$ := 0;
            for   $n$ = 1 to 2 do begin
               $w_p(n) := \hat{W}_{ap}(n)^{[1]} W_p(n)$;   $X := X + w_p(n)\, Y_{pa}(n)$                (5.2b)
            end;
            for   $n$ = 1 to 2 do   if $X \neq 0$   then   $W_a(n) := W_a(n) + w_p(n)$            (5.3b)
         end;
         for each $q$ in A($a$) do      for $n$ = 1 to 2 do    $Z := Z + w_q(n)Y_{dq}(n)$ ;         (5.4b)
         $\hat{V}_{ad} := \hat{V}_{ad} + Z/2$ ;      if   $Z \neq 0$ then   $flag$ := 1;
         for $n$ = 1 to 2 do  $\tilde{V}_{ad}(n) := \tilde{V}_{ad}(n) + W_a(n)$                       (5.6b)
      end
   end
end;
```

Step 2: Calculation of Link Flows by OD-pairs

```
for each $(o,d)$ in $L^O \times L^D$ do begin
    $\hat{V}_{od} := \hat{V}_{ad}$ ,where $a$ is a link connected with link $o$;
    for each $a$ in $L$ do  $x_a^{od} := \frac{1}{2} q_{od} \sum_{n=1}^{2} \tilde{V}_{oa}(n)\, \tilde{V}_{ad}(n)\, Y_{do}(n) \big/ \hat{V}_{od}$ ;      (5.7)
end;
```

5.3.Extension to the Flow Dependent (Equilibrium) Case

We can extend the model to the flow dependent case (i.e. stochastic equilibrium model) where the link travel time is given by $t_a = t_a(x_a)$. The simple strategy for developing the algorithm is to employ the method of successive averages (cf. **POWELL and SHEFFI** (1982), **SHEFFI and POWELL**(1982)). It can be summarized as follows:

Step 0: Initialization

(a) Set iteration counter: $k := 1$.

(b) Find an initial feasible link flow pattern $\{x_a^{(1)}\}$.

Step 1: Direction Finding

(a) Set the link cost vector $\mathbf{t}^{(k)}$: $t_a^{(k)} := t_a(x_a^{(k)}) \qquad \forall a \in L$

(b) Calculate the auxiliary link flow pattern $\{y_a^{(k)}\}$ by the stochastic assignment for the fixed link cost pattern $\{t_a^{(k)}\}$.

(c) Set the direction vector $\mathbf{d}^{(k)}$: $d_a^{(k)} := y_a^{(k)} - x_a^{(k)} \qquad \forall a \in L$

Step 2: Link Flow Update

Update the link flow pattern $\mathbf{x}^{(k+1)}$: $x_a^{(k+1)} := x_a^{(k)} + a^{(k)} d_a^{(k)} \qquad \forall a \in L$,

where $a^{(k)}$ is a real number such that

$$0 < a^{(k)} < 1, \quad \sum_{k=1}^{\infty} a^{(k)} = +\infty, \text{ and } \sum_{k=1}^{\infty} (a^{(k)})^2 < \infty .$$

Step 3: Convergence Test

If appropriate convergence criteria are satisfied then stop;
otherwise set $k := k+1$ and go to Step 1.

Obviously, Step 1(b) can be efficiently accomplished by the procedure presented in 5.2. It is noteworthy that we can not guarantee the convergence if we employ the Dial's algorithm instead of our procedure. The reason is that the path set generated by the Dial's algorithm varies between iterations due to the updating of the link cost pattern, which prevents us to guarantee the closedness (see **ZANGWILL** (1969)) of the algorithmic mapping. On the other hand, our procedure does not suffer from such troubles, since the path set is fixed whatever link cost patterns. Thus, we obtained the algorithm that is guaranteed to converge to the stochastic equilibrium flow pattern.

For the development of more efficient algorithms, it is convenient to exploit the equivalent optimization problem for the model. Conventionally, the problem of **Fisk**(1980) is well known as the equivalent programming for a standard stochastic equilibrium assignment, and therefore, we may construct the similar the programming for our model. Such an approach, however, can not exploit the programming as the convenient tool for developing efficient algorithms. The reason is that the objective function is hard to evaluate due to the explicit inclusion of path flows as in the Fisk's problem. Thus, it is natural to seek for a more convenient programming that is formulated by only link variables. For the assignment model shown in Section 1, **Akamatsu** (1996b,c) showed the following link-based equivalent problem:

[P1] $$\min. \sum_{a\in L} t_a x_a + \frac{1}{q}\sum_{o\in O}\left(\sum_{a\in L} x_a^o \ln x_a^o - \sum_{k\in N} y_k^o \ln y_k^o\right) \tag{5.8}$$

subject to

$$x_a = \sum_{o\in O} x_a^o \qquad \forall a \in L \tag{5.9}$$

$$y_k^o = \sum_{a\in In(k)} x_a^o \qquad \forall k \in N,\ \forall o \in O$$

(5.10)

$$\sum_{a\in In(k)} x_a^o - \sum_{a\in Out(k)} x_a^o - q_{ok} = 0 \qquad \forall k \in N, k \neq o,\ \forall o \in O$$

(5.11)

$$x_a^o \geq 0 \qquad \forall a \in L, \qquad \forall o \in O$$

(5.12)

where *In*(*k*) denotes the set of links entering into node *k*, and *Out*(*k*) the set of links emanating from node *k*. He also showed that the flow dependent case yields

[P2] $$\min. \sum_{a\in L}\int_0^{x_a} t_a(w)dw + \frac{1}{q}\sum_{o\in O}\left(\sum_{a\in L} x_a^o \ln x_a^o - \sum_{k\in N} y_k^o \ln y_k^o\right)$$

(5.13)

subject to (5.9)-(5-12).

Similarly, we see that the flow dependent version of the model in Section 3 is equivalent to the following problem:

[P3] $$\min. \sum_{a\in L}\int_0^{x_a} t_a(w)dw + \sum_{a\in L}\sum_{b\in A(a)} |w_{ab}|\ x_a + \frac{1}{q}\sum_{o\in O}\left(\sum_{a\in L} x_a^o \ln x_a^o - \sum_{k\in N} y_k^o \ln y_k^o\right)$$

(5.14)

subject to (5.9)-(5-12).

As for the model in Section 4, however, finding the equivalent programming is not a trivial problem. A naive conjecture is that the programming [P3] is equivalent if we regard that the feasible region constituted by (5.9)-(5.12) includes no cycles. Although the conjecture is simple, the proof seems to be somewhat difficult. Regrettably, I must confess that I have not yet obtained the clear answer to this problem, and it still remains to be studied.

6. Conclusions and Future Research

In this paper we have developed the efficient method for the LOGIT based stochastic (equilibrium) assignment model whose path set consists of only simple paths without any restrictions. To achieve the purpose, the novel technique that represents the network structure by means of the *complex* "impedance" (adjacency) matrices was introduced. The technique combined with the Markov chain assignment (originally developed by **Sasaki** (1965), **Bell** (1995) and **Akamatsu**(1996a)) enabled us to completely eliminate the cycle flows from the path set for the assignment without explicit path enumeration.

For the future research opportunity, it may be interesting to explore the applications of the technique to various other problems in the graph / network theory. The reason why we suggest the exploration is that the technique is fairly natural from the mathematical view point. As is well known, almost all the contents of the basic graph theory parallels the linear algebra. For example, the various graph / network problems called "path problem" have the same algebraic structure with the system of linear equations (see, for example, **Carre** (1979), **Takenaka** (1989) etc.). Recall here that one of the most significant properties of the linear mapping is "enlargement and rotation". Obviously, it is essential to analyze the property in not a real number space but a complex number space. Nevertheless, the mathematical means to represent / analyze the graph structure in the conventional graph theory have been restricted to the real number matrices such as incidence matrices or adjacency matrices with zero / one elements; the "rotation" seems to have been neglected in the theory while the "enlargement" has the correspondence in the theory. Thus, there seems to be some significant problems that are natural to be analyzed in complex number space; the classification of graphs by means of the spectra of the adjacency matrices (see, for example, **Cvetkovic** et al. (1978), **Takenaka** (1989) etc.) is such an example that can be meaningfully extended to the complex case.

For another important future research, we should mention the exploration of the equivalent programming problem for the LOGIT based stochastic assignment *whose path set consists of only simple paths*. While the derivation of the path-based problem is very easy, the *link-based problem* is an open question at present. The link-based programming is indispensable for the development of the efficient algorithms for the flow dependent case (i.e. the Stochastic User Equilibrium assignment), and therefore, the construction of the programming is an important problem to be solved. We hope this study gives the foundation for the construction.

REFERENCES

T. Akamatsu, “Integrated Transportation Network Models based on Stochastic Equilibrium Approach” (in Japanese), Dr. course Dissertation, Department of Civil Engineering, Tokyo University, Tokyo, 1990.

T. Akamatsu, “Cyclic Flows, Markov Process and Stochastic Traffic Assignment”, to appear in *Transportation Research Part B* (1996a).

T. Akamatsu, “Decomposition of Path Choice Entropy in General Transport Networks”, to appear in *Transportation Science* (1996b).

T. Akamatsu, “Partial Linearization Algorithm for Stochastic Equilibrium Assignment without Path Restrictions”, submitted to *Transportation Research Part B* (1996c).

T. Akamatsu and Y. Matsumoto, "A Stochastic User Equilibrium Model with Elastic Demand and Its Solution Method" (in Japanese), *Proc. of JSCE* **401**, pp.109-118 (1989).

M.G.H. Bell, “Alternatives to Dial's Logit Assignment Algorithm”, *Transportation Research* **29B**, pp.287-295 (1995).

J.E. Burrell, “Multipath Route Assignment and Its Applications to Capacity Restraint”, *Proc. of the 4th International Symposium on the Theory of Road Traffic Flow*, West Germany, 1968.

J.E. Burrell, “Multipath Route Assignment: A Comparison of Two Methods”, In M. Florian (Ed.) *Traffic Equilibrium Methods*, Lecture Notes in Economics and Mathematical Systems 118, Springer-Verlag, New York, pp.229-239, 1976.

B. Carre, *Graphs and Networks*, Oxford University Press, 1979.

D.M. Cvetkovic, M.Doob and H.Sachs, *Spectra of Graphs*, Academic Press, 1978.

C.F. Daganzo, “Unconstrained Extremal Formulation of Some Transportation Equilibrium Problem”, *Transportation Science* **16**, pp.332-360 (1982).

C.F. Daganzo, “Stochastic Network Equilibrium with Multiple Vehicle Types and Asymmetric Indefinite Link Cost Jacobian”, *Transportation Science* **17**, pp.282-300 (1983).

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

R.B. Dial, “A Probabilistic Multipath Traffic Assignment Algorithm which Obviates Path Enumeration”, *Transportation Research* **5**, pp.83-111 (1971).

C.S. Fisk, "Some Developments in Equilibrium Traffic Assignment", *Transportation Research* **14B**, pp.243-255 (1980).

F.M. Leurent, “Contribution to Logit Assignment Model”, *Transportation Research Record* **1493**, pp.207-212 (1996).

P. Mirchandani and H. Soroush, “General Traffic Equilibrium with Probabilistic Travel Time and Perceptions”, *Transportation Science* **21**, pp.133-152 (1987).

T. Miyagi, “On the Formulation of a Stochastic User Equilibrium Model Consistent with Random Utility Theory”, *Proc. of the 4th World Conference for Transportation Research*, pp.1619-1635, 1985.

W.B. Powell and Y. Sheffi, “The Convergence of Equilibrium Algorithms with Predtermined Step Sizes”, *Transportation Science* **16**, pp.45-55 (1982).

T. Sasaki, “Theory of Traffic Assignment through Absorbing Markov Process” (in Japanese), *Proc. of JSCE* **121**, pp.28-32 (1965).

Y. Sheffi and C.F. Daganzo, “Computation of Equilibrium over Transportation Networks: The Case of Disaggregate Demand Models”, *Transportation Science* **14**, pp.155-173 (1980).

Y. Sheffi and W.B. Powell, “A Comparison of Stochastic and Deterministic Traffic Assignment over Congested Networks”, *Transportation Research* **15B**, pp.53-64 (1981).

Y. Sheffi and W.B. Powell, “An Algorithm for the Equilibrium Assignment Problem with Random Link Times”, *Networks* **12**, pp.191-207 (1982).

T. Takenaka, *Linear Algebraic Graph Theory* (in Japanese), Baifukan, 1989.

R.L. Tobin, “An Extension of Dial's Algorithm Utilizing a Model of Tripmaker's Perceptions”, *Transportation Research* **11**, pp.337-342 (1977).

D.Van Vliet, “Selected Node-Pair Analysis in Dial's Assignment Algorithm”, *Transportation Research* **15B**, pp.65-68 (1981).

D.Van. Vliet and P.D.C. Dow, “Capacity-restrained Road Assignment”, *Traffic Engineering & Control*, pp.296-305 (1979).

H.Von Falkenhausen, “Traffic Assignment by a Stochastic Model”, *Proc. of International Conference on Operational Science*, pp.415-421, 1966.

W.I. Zangwill, *Nonlinear Programming: A Unified Approach*, Prentice-Hall, 1969.

R.J. Wilson, *Introduction to Graph Theory*, Longman, 1985.