PERGAMON

Transportation Research Part B 37 (2003) 225–249

TRANSPORTATION RESEARCH PART B

www.elsevier.com/locate/trb

Dynamic traffic modelling and dynamic stochastic user equilibrium assignment for general road networks

Sangjin Han *

[a] *Centre for Transport Studies, UCL, Gower Street, London WC1E 6BT, UK*

Received 1 November 2000; received in revised form 28 December 2001; accepted 6 January 2002

Abstract

This paper investigates the requirements of dynamic traffic modelling, and proposes the deterministic queuing model as a plausible link performance function to describe the relationship between inflows, outflows, and link travel costs in time-varying condition. Then, it explains how we can perform logit-based stochastic network loading for general road networks in the dynamic case. In particular, this paper shows how to perform dynamic stochastic network loadings for many-to-many origin–destination pairs, and what should be considered to maintain correct flow propagation in the network loading process. Next, this paper shows how the stochastic dynamic user equilibrium (SDUE) assignment problem can be solved without direct evaluation of the objective function. For this purpose, a quadratic interpolation, the method of successive average, and the pure network loading method are adopted at the line-search step in the solution algorithm. Numerical examples show that the present SDUE assignment model with a quadratic interpolation gives rise to a convergent solution with good quality whilst needing less computation time. Furthermore, it is found that the predictive cost-flow association (Proceedings of the European Transport Conferences, Seminar F, P434, 1999, p. 79) is preferable to the reactive one because the former can produce consistent assignment patterns regardless of the size of dispersion parameter θ in the logit model for route choice.

Keywords: Dynamic traffic modelling; Flow propagation; Stochastic dynamic user equilibrium assignment; Predictive or reactive cost-flow association; Line-search method

* Present address: Korea Transport Institute, 2311 Daehwadong, Ilsangu, Koyangsi 411 701, Korea. Tel.: +82-31-910-3112; fax: +82-31-910-3235.

E-mail address: han@koti.re.kr (S. Han).

0191-2615/02/$ - see front matter
PII: S0191-2615(02)00009-7

1. Introduction

The outstanding advantage of stochastic assignment compared to its deterministic counterpart is that it models how travellers behave in networks with less strict assumptions. In particular, it allows for the possibility that travellers may have either or both imperfect information and different perceptions toward travel costs rather than perfect information and homogeneous perceptions. Consequently, there has been a substantial effort to apply this stochastic assignment principle to time-varying networks.

For example, Ben-Akiva et al. (1986) extended the stochastic equilibrium model (De Palma et al., 1983) to a day-to-day dynamic case using a deterministic queuing model for a link performance function. However, their model was applied only to networks with a single origin–destination pair and non-overlapping routes. Later, Vythoulkas (1990) applied Ben-Akiva et al.'s (1986) model to a general network. However, he used a link performance function, which does not always maintain the first-in-first-out (FIFO) discipline and an equilibrium solution is not always guaranteed because the model is based on simulation. On the other hand, Cascetta and Cantarella (1991) modelled both day-to-day and within-day dynamic assignments using a stochastic process, and applied them to a general network. Recently, Ran and Boyce (1996) improved Dial's (1971) logit-based STOCH network loading method, so that it could be applied to a time-varying network (it is named as dynamic stochastic network loading method (DYNASTOCH)). However, they did not apply their model to a general network, and also the link performance function that they adopted is not plausible in terms of the FIFO discipline.

In this study, we first investigate what should be considered in dynamic traffic modelling in order to describe the relationship correctly between traffic inflows, outflows, and travel costs. We suggest the deterministic queuing model for a link performance function because it can maintain all requirements for a dynamic traffic modelling. Then, a dynamic generalisation of a stochastic user equilibrium (SUE) state is formulated as a variational inequality following Ran and Boyce (1996). In particular, this paper explains how we can find the stochastic dynamic user equilibrium (SDUE) assignment without direct evaluation of the objective function when we solve the variational inequality in the framework of the diagonalisation method. It also explains in detail how to perform a stochastic dynamic network loading whilst maintaining correct flow propagation for general networks which have more than a single origin–destination pair. Then numerical examples of a SDUE assignment in test networks are given when we adopt the deterministic queuing model for a link performance function and the realistic (or ideal) cost concept for a route travel cost.

Section 2 explains what is required in the dynamic traffic modelling and suggests the deterministic queuing model for a plausible link performance function. The formulation of SDUE assignment problem will be shown in Section 3. Section 4 describes a solution algorithm for the SDUE assignment problem together with a stochastic dynamic network loading method. Application of the developed SDUE assignment to example networks is shown in Section 5.

2. Requirements in dynamic traffic modelling

In order to perform dynamic traffic assignment properly, we need to consider some requirements such as causality, the FIFO discipline and satisfactory flow propagation when we model traffic inflows, outflows, and travel costs on a link. Moreover, the way of calculating route travel cost is not so straightforward as in the static case because travel cost can change over the time during which travellers traverse a route. Lastly, when we perform dynamic traffic assignment in discrete rather than continuous time, correct association of costs with flows for each time increment becomes crucial.

2.1. Causality

Causality suggests that the current travellers' behaviour is influenced by that of others in the past rather than by that in the future in a time-varying condition (Heydecker and Addison, 1996). This seems to be a natural requirement considering that travellers decide their travel pattern based on the network conditions that they will encounter which have arisen as a result of past events rather than future ones. Thus, causality can be maintained along a trajectory when traffic behaves depending on the downstream conditions that have been formed ahead of them rather than on the upstream conditions which will be formed after them. Accordingly, this downstream dependent condition should be reflected in any link cost function in order for it to be used for the dynamic assignment. Furthermore, causality explains that link costs in dynamic assignments can be affected by flows at previous times as well as by those at the current time, but it is not affected by those at future times.

2.2. First-in-first-out discipline

The FIFO discipline is observed implicitly in static assignment, because it is assumed that travel times are identical for all travellers on the same route. This discipline should be observed in a dynamic assignment, since we do not generally expect that travellers arrive at the destination earlier than those who departed the origin before them. The FIFO discipline can be written as follows for a link:

$$\frac{\mathrm{d}\tau_a(t)}{\mathrm{d}t} \geqslant 0 \tag{1}$$

where, $\tau_a(t)$ denotes the exit time from link a associated with entering it at time t.

Condition (1) states that the exit time, $\tau_a(t)$, should increase as the departure time t increases, so that overtaking cannot occur on the link. This FIFO condition can be found in several places in the literature such as Friesz et al. (1993), Astarita (1996), and Ran and Boyce (1996, p. 79).

2.3. Flow propagation

The flow propagation represents how the flow rate varies along a vehicle trajectory. In a static assignment, this is not considered explicitly because the flows on a route are assumed to remain constant along it. However, in a dynamic assignment, flow rates can change along a route according to the network conditions. For instance, if there is a bottleneck at a certain point along the route, the flows which enter the queue for the bottleneck can be arbitrarily large whereas the outflows from the bottleneck are limited by its capacity. This means that the flow rates can change along a route and over time. Generally, the flow propagation which follows the FIFO discipline can be written as:

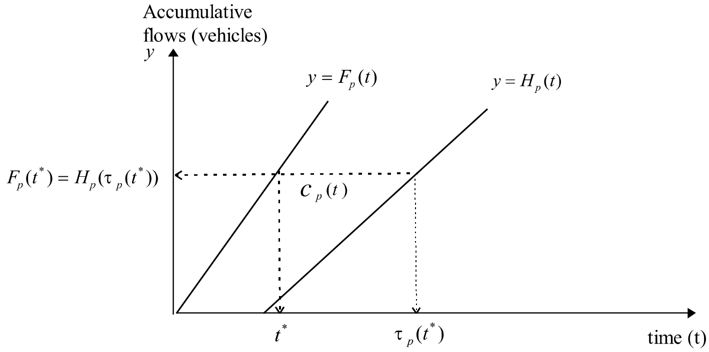

Fig. 1. Cumulative flows and travel time.

$$h_p(\tau_p(t)) = \frac{f_p(t)}{\mathrm{d}\tau_p(t)/\mathrm{d}t} \tag{2}$$

where, $f_p(t)$ is inflow to route p at time t, $h_p(t)$ is outflow from route p at time t, $\tau_p(t)$ is exit time from route p associated with entering it at time t.

Eq. (2) states that the change rate of exit time and the inflow rate can explain the outflow rate at the downstream end. Therefore, we can discover how the flow rate changes along a route over time according to (2). This can be derived as follows:

In general, the cumulative inflows $F_p(t)$ for route p at time t, would be the same as the cumulative outflows $H_p(\tau_p(t))$ from route p at the exit time $\tau_p(t)$, if the FIFO discipline is to hold. Fig. 1 illustrates this relationship.

Accordingly, we can derive the following relationship:

$$F_p(t) = H_p(\tau_p(t)) \tag{3}$$

or

$$\int_{u=0}^{t} f_p(u)\,\mathrm{d}u = \int_{w=0}^{\tau_p(t)} h_p(w)\,\mathrm{d}w \tag{4}$$

If we differentiate (4), and use the chain rule,

$$f_p(t) = h_p(\tau_p(t))\frac{\mathrm{d}\tau_p(t)}{\mathrm{d}t} \tag{5}$$

Naturally, flow propagation condition (2) can hold for links because Eq. (3) holds for a link as well. The flow propagation can be expressed for a link as:

$$g_a(\tau_a(t)) = \frac{e_a(t)}{\mathrm{d}\tau_a(t)/\mathrm{d}t} \tag{6}$$

where, $e_a(t)$ is the inflows to link a at time t, $g_a(t)$ is the outflows from link a at time t, $\tau_a(t)$ is an exit time from link a associated with entering it at time t.

2.4. Route travel cost

In static assignments, route travel costs are calculated by summing link travel costs along the links that constitute a route because travel cost is represented as a constant regardless of time.

However, in the dynamic case, link flows and resulting link travel costs are not necessarily constant over time. That means the network condition can vary as travellers traverse a route.

In this context, Ran and Boyce (1996) suggested two travel cost concepts such as 'ideal' and 'instantaneous' route travel costs in dynamic assignments. The 'ideal' or 'realistic' route travel cost is the one that a traveller will experience whilst traversing the network. In other words, when we calculate a route travel cost, we take account of the fact that it takes some time to exit a link after entering it, and the network conditions may have changed by that time. Accordingly, the cost on the next link is calculated based on the condition at the time of entry to the link. On the other hand, when we calculate the 'instantaneous' or 'naive' travel cost, we assume that current network conditions will not change substantially until travellers arrive at their destinations. Therefore, the instantaneous travel cost is calculated by summing the travel cost on each of the links that form a route at the time of entry to the first link.

We can express 'realistic' and 'naive' route travel costs in mathematical terms as follows: Note that the equation for naive route travel cost (7) has a recursive structure.

$$C_p^R(t) = c_{a_1}(t) + c_{a_2}(t + c_{a_1}(t)) + \cdots + c_{a_m}(t + c_{a_1}(t) + c_{a_2}(t + c_{a_1}(t)) + \cdots + c_{a_{m-1}}) \tag{7}$$

$$C_p^N(t) = c_{a_1}(t) + c_{a_2}(t) + \cdots + c_{a_m}(t) \tag{8}$$

where, $C_p^R(t)$ is the realistic travel cost for route p at time t, $C_p^N(t)$ is the naive travel cost for route p at time t, $c_a(t)$ is the cost of link a at time t and $a_1, a_2, \dots, a_m \in p$.

We note, however, that dynamic route travel cost concepts are referred to using various terms according to the literature. For example, Friesz et al. (1993) first distinguished the realistic travel cost concept from the naive counterpart, and called the assignment model which adopted the naive travel cost concept the Boston Traffic Equilibrium assignment, and that which adopted the realistic travel cost concept the Path Integral Equilibrium assignment. Similarly, Buisson et al. (1998), and Kuwahara and Akamatsu (1997) called realistic and naive travel cost concepts, 'predictive' and 'reactive' travel costs respectively.

2.5. Cost-flow association in discrete time

When representing time as a discrete variable, we need to consider how to associate the flows and resulting link travel costs (or times) in a time increment, because they do not necessarily correspond to each other at the same time instant. For example, the present flow will affect the cost incurred only by the travellers who come after the present time instant according to the principle of causality. Therefore, the cost which corresponds to the flow during the time increment, $[t, t + \Delta t)$, can be found at the final instant $t + \Delta t$ rather than at the initial instant t. In this respect, if we assign traffic during a certain time increment based on the resulting costs (or the costs at the final instant in a time increment), it is called a 'predictive' assignment, whereas if we assign them based on the costs from the previous flow pattern (or the costs at the initial instant in a time increment), it is called a 'reactive' assignment. These different assignment concepts in discrete times are depicted in Fig. 2.

In Fig. 2, $c_a(t)$ denotes the cost incurred by entering link a at instant t, whilst $e_a(t)$ denotes the flow into link a throughout a time increment $[t, t + \Delta t)$. This figure shows that $e_a(t)$ is associated with $c_a(t + \Delta t)$ (or the costs at the final instant in a time increment) in the predictive assignment,

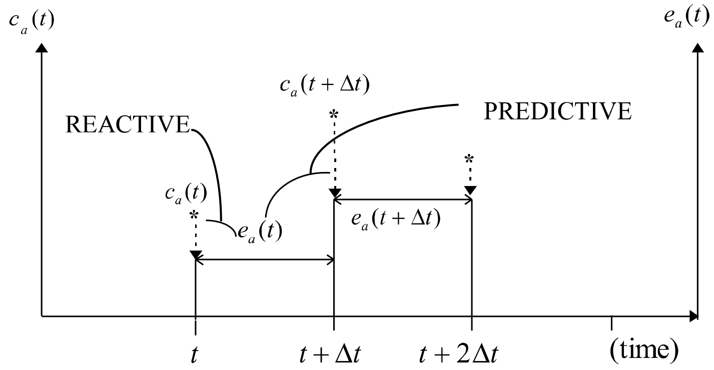

Fig. 2. Difference between predictive and reactive assignment.

whereas it is associated with $c_a(t)$ (or the costs at the initial instant in a time increment) in the reactive assignment.

Although the difference between the predictive and reactive concepts of the assignment does not arise in the continuous time case, it becomes important for any finite discrete time increment. Thus, Heydecker and Verlander (1999) showed that the user equilibrium solution from predictive assignment is quite different from that from reactive assignment.

3. Formulation of the model

3.1. Link performance function

According to the deterministic queuing model (Newell, 1982), traffic travels freely along links and then possibly incurs delay at the downstream end if there is a queue. While inflow is less than the capacity of the link and there is no queue, the outflow is the same as the inflow, otherwise the outflow is equal to capacity. We can formulate the deterministic queuing model in mathematical terms as:

$$\frac{\mathrm{d}L}{\mathrm{d}t} = \begin{cases} 0 & \text{if } (L_a(t) = 0,\ e_a(t-\phi_a) < Q_a) \\ e_a(t-\phi_a) - Q_a & \text{otherwise} \end{cases} \tag{9a}$$

$$g_a(t) = \begin{cases} e_a(t-\phi_a) & \text{if } (L_a(t) = 0,\ e_a(t-\phi_a) < Q_a) \\ Q_a & \text{otherwise} \end{cases} \tag{9b}$$

$$d_a(t) = \frac{L_a(t+\phi_a)}{Q_a} \tag{9c}$$

$$c_a(t) = \phi_a + d_a(t) \tag{9d}$$

where, $L_a(t)$ is the queue length on link a at time t, $g_a(\tau(t))$ is the outflows from link a at the exit time $\tau_a(t)$, $c_a(t)$ is the travel cost on link a at time t, $d_a(t)$ is the delay incurred by a vehicle that enters link a at time t, ϕ_a is the free-flow travel cost for link a, Q_a is the capacity of link a.

This deterministic queuing model guarantees the FIFO discipline and flow propagation because the necessary conditions for FIFO discipline (1) and flow propagation (6) are met implicitly. We can also note that outflows and link costs are calculated based on the queue length at the downstream end from (9b)–(9d). This shows that this link performance function satisfies causality because these queue lengths depend only on earlier flows. Furthermore, the model is simple to calculate from Eqs. (9a)–(9d). However, unless the inflow has exceeded capacity so that there is a non-zero queue, travel cost is constant at free-flow travel costs. This means that travel cost is independent of the flow in this range. In this respect, the deterministic queuing model does not represent an entirely plausible relationship between flows and travel costs (times) if the inflows are less than capacity and there is no queue.

We can see that the deterministic queuing model meets the FIFO condition (1) by (10) because $\mathrm{d}L_a(t+\phi_a)/\mathrm{d}t$ cannot be less than $-Q_a$ from (9a):

$$\tau_a(t) = t + c_a(t) = t + \phi_a + \frac{L_a(t+\phi_a)}{Q_a} \tag{10}$$

$$\frac{\mathrm{d}\tau_a(t)}{\mathrm{d}t} = 1 + \frac{1}{Q_a}\frac{\mathrm{d}L_a(t+\phi_a)}{\mathrm{d}t} \geqslant 0 \tag{11}$$

In addition, we can show that the flow propagation (6) holds because;

$$g_a(\tau_a(t)) = \frac{e_a(t)}{\frac{\mathrm{d}\tau_a(t)}{\mathrm{d}t}} = \frac{e_a(t)}{1+\frac{1}{Q_a}\frac{\mathrm{d}L_a(t+\phi_a)}{\mathrm{d}t}} = \begin{cases} e_a(t) & \text{if } (L_a(t+\phi_a)=0,\ e_a(t) < Q_a) \\ Q_a & \text{otherwise} \end{cases} \tag{12}$$

The condition (12) corresponds to the outflow function (9b). This means that the flow propagation (6) holds implicitly in the deterministic queuing model.

Unlike the deterministic queuing model, some link performance functions which calculate outflows on the basis of the number of vehicles over the whole link cannot maintain causality, the FIFO discipline, and correct flow propagation (see, for example, Astarita, 1996; Heydecker and Addison, 1998; Han, 2000). Apart from the deterministic queuing model, there are link performance functions that can maintain those requirements. For example, there are a linear function form (Astarita, 1996) and the kinematic wave model (Heydecker and Addison, 1996).

3.2. Route choice model

In stochastic route choice, not all travellers use a minimum cost route as in the deterministic counterpart, hence it is assumed that there are perception differences between travellers for the minimum cost route. Because of perception difference between travellers, they believe that they use minimum cost routes, although not all routes in use are minimum cost one according to an objective measure.

Traditionally, there have been two kinds of stochastic route choice model for traffic assignment problems. One is a logit, and the other is a probit model. In the present SDUE assignment model, logit model is used based on Dial (1971). According to the logit model, a route choice probability in the time-varying network can be decided by following equation:

$$P_p^{od}(t) = \frac{\exp(-\theta C_p^{od}(t+\Delta t))}{\sum_{q\in R_{od}} \exp(-\theta C_q^{od}(t+\Delta t))} \tag{13}$$

where, $P_p^{od}(t)$ is a route choice probability for route p for origin–destination pair od at time t, $C_p^{od}(t)$ is a travel time (cost) for route p for an origin–destination pair od at time t, θ is a dispersion parameter, R_{od} is the route set for origin–destination pair od. Note that the cost term $C_p^{od}(t)$ in Eq. (13) should be evaluated at instant $t + \Delta t$ rather than instant t in the predictive cost-flow association if we deal with the route choice problem in discrete time.

3.3. Formulation of the stochastic dynamic user equilibrium

We can define dynamic SDUE principle as follows: This is a dynamic generalisation of the (static) SUE principle (Daganzo and Sheffi, 1977).

"At each instant no traveller believes that he or she can improve his or her perceived travel cost by unilaterally changing route".

The above principle can be written mathematically according to Sheffi and Powell (1982) as:

$$P_p^{od}(t) = \frac{f_p^{od}(t)}{q^{od}(t)} \quad p \in R_{od}, \ \forall od \tag{14a}$$

$$\sum_{p \in R_{od}} f_p^{od}(t) = q^{od}(t) \quad \forall od \tag{14b}$$

$$f_p^{od}(t) \geqslant 0 \quad \forall od \tag{14c}$$

where,

$$P_p(t) = \Pr(\hat{C}_p(t) \leqslant \hat{C}_h(t) \quad \forall h \in R_{od} | \mathbf{C}(t)) \quad \forall p \in R_{od} \tag{15}$$

$q^{od}(t)$ is the traffic demand from origin o to destination d at time t, $f_p^{od}(t)$ is the flow on route p connecting origin o to destination d at time t.

The route choice probability $P_p^{od}(t)$ in (14a) is the one that the perceived travel cost on route p, $\hat{C}_p(t)$ is the smallest of all routes for origin–destination pair od at instant t and it is dependent on the route cost pattern at instant t, $\mathbf{C}(t)$. In the SDUE state, the route choice probability which is obtained from (14a)–(14c) should be identical to the one from (15). This condition can be achieved when the perceived cost pattern does not change following the corresponding dynamic stochastic network loading.

3.4. Variational inequality

The above SDUE state was first formulated as a variational inequality by Ran and Boyce (1996, pp. 190–193)as:

The demand feasibility condition for route flows $\mathbf{f}(\cdot)$ is assured if,

$$\begin{aligned} &f_p(t) \geqslant 0 \qquad \forall p \in R_{od}, \ \forall t \\ &\sum_{p \in R_{od}} f_p(t) = q^{od}(t) \quad \forall t \end{aligned} \tag{16}$$

Then an assignment at time t, expressed in the form of route inflows $f_p^*(t)$ is an equilibrium if and only if:

$$\sum_{od}\sum_{p} \mathrm{K}_p^{od}(t)\{f_p^{od}(t) - f_p^{od^*}(t)\} \geqslant 0 \quad \forall \mathbf{f} \in D \tag{17}$$

In Eq. (17), the cost term $\mathrm{K}_p^{od}(t)$ is defined as follows:

$$\mathrm{K}_p^{od}(t) = \{f_p^{od}(t) - q^{od}(t)P_p^{od}(t)\}\frac{\partial C_p^{od}(t)}{\partial f_p^{od}(t)} \tag{18}$$

For more detailed explanation, refer to Nagurney (1993) and Ran and Boyce (1996). They proved that the variational inequality (17) holds in the SDUE state, by showing that the right-hand side of Eq. (18) becomes zero in the SDUE state once we assume that route travel costs $C_p^{od}(t)$ increases as the associated route flows $f_p^{od}(t)$ increases, i.e.,

$$\frac{\partial C_p^{od}(t)}{\partial f_p^{od}(t)} \geqslant 0 \tag{19}$$

4. Solution algorithm

The optimal solution to variational inequality (17) can be found by solving the equivalent mathematical program for a separable link cost function in the framework of the diagonalisation method (Ran and Boyce, 1996). Before describing the diagonalisation method in detail, we will see how we can perform stochastic dynamic network loading, which is essential to find feasible link flow patterns. Note that this paper particularly develops an existing stochastic dynamic network loading method so that it can be applicable to general road networks.

4.1. Dynamic stochastic network loading method

In the case of the static logit-based stochastic assignment, we can use efficient network loading methods such as Dial's (1971) STOCH algorithm which do not need path enumeration. Similarly, we can apply this efficient network loading method to the dynamic case. In particular, Ran and Boyce (1996) proposed the DYNASTOCH algorithm for stochastic dynamic network loading by developing the STOCH algorithm. This algorithm maintains the structure of the original STOCH algorithm, so only deals with reasonable routes, and assigns traffic according to the link choice probability in a forward pass after calculating link weights in a backward pass. The DYNASTOCH algorithm can be summarised for each origin–destination pair for the realistic route travel cost in the predictive cost-flow association as follows:

Note that, unlike the original STOCH algorithm, traffic is assigned in the forward pass rather than the backward pass in order to maintain correct flow propagation:

Step 0: Calculation of link likelihood [1]

Compute the minimum ideal travel cost $C_{jd}^*(t)$ for travellers departing node j during time increment t. Calculate the likelihood, $a_{(i,j)}(t)$, for each link (i,j) during each time increment t:

[1] The term 'likelihood' in here does not accord with that in statistics. It refers to how strongly links are associated with reasonable routes.

$$a_{(i,j)}(t) = \begin{cases} \exp\{\theta[C^*_{id}(t) - C^*_{jd}(t + c_{(i,j)}(t)) - c_{(i,j)}(t)]\} & \text{if } C^{id}_o > C^{jd}_o \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

where, $C^*_{jd}(t)$ is the minimum ideal travel cost from j to d by departing the node j at time t, C^{id}_o is the ideal travel cost from i to d when there is no flow in the network

Step 1: Backward pass

By examining all nodes j from the destination d in ascending sequence with respect to $C^*_{jd}(t)$, calculate $w_{(i,j)}(t)$, the link weight for each link (i,j) during each time increment t:

$$w_{(i,j)}(t) = \begin{cases} a_{(i,j)}(t) & \text{if } j = d \\ a_{(i,j)}(t) \sum_{(j,k)\in A(j)} w_{(j,k)}[t + c_{(i,j)}(t)] & \text{otherwise} \end{cases} \tag{21}$$

where, $A(j)$ is the set of links starting from node j.

When the origin node o is reached, stop

Step 2: Forward pass

Consider all nodes i in descending sequence with respect to $C^*_{id}(t)$, starting with the origin o. When each node i is considered during each time increment t, compute the inflow to each link (i,j) during each time increment t using the following formula:

$$e_{(i,j)}(t) = \begin{cases} q^{od}(t) \dfrac{w_{(i,j)}(t+\Delta t)}{\sum\limits_{(i,k)\in A(i)} w_{(i,k)}(t+\Delta t)} & \text{if } i = o \\ \left\{ \sum\limits_{(k,i)\in B(i)} g_{(k,i)}(t) \right\} \dfrac{w_{(i,j)}(t+\Delta t)}{\sum\limits_{(i,k)\in A(i)} w_{(i,k)}(t+\Delta t)} & \text{otherwise} \end{cases} \tag{22}$$

where, $B(i)$ is the set of links ending at node i.

The DYNASTOCH algorithm has been explained only for a special case when there is one origin–destination pair. However, this algorithm can be developed for general cases when there is more than one origin–destination pair as Fig. 3. This is one of crucial developments, which are made by the present study.

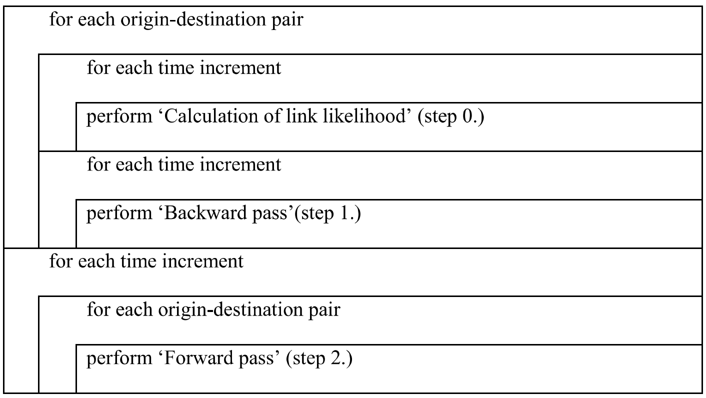

Fig. 3. DYNASTOCH algorithm for many-to-many case.

In Fig. 3, the forward pass is performed for all origin–destination pairs at each time increment, after finishing the backward pass for all origin–destination pairs and for all time increments. In fact, the backward pass and the forward pass cannot be calculated under the same time increment. This results from the different order of considering time increments between the backward pass and the forward pass. For example, in the backward pass, the link weight at time t cannot be calculated unless the future weights of connected links are known. That means link weights should be calculated from a destination node in descending order from the time when the last traveller exits the network (last time) to the time when the first traveller enters the network (initial time). In contrast, the forward pass should be started from the origin node and performed from the initial time to the last time in order to maintain correct flow propagation. Therefore, this conflicting feature between the backward pass and the forward pass requires the storage of link weights for all origin–destination pairs and for all time increments in the backward pass beforehand, in order to perform stochastic dynamic network loading properly in the forward pass.

Apart from the process of the DYNASTOCH algorithm, there are two more points which should be noted. One is the necessity of fixing and respecting the reasonable route set, and the other is the need for interpolation in calculating link weights.

First, the reasonable routes in which each link takes travellers further from the origin node rather than closer to it must be fixed in order to produce a convergent solution when we apply the DYNASTOCH algorithm for SDUE assignment. Otherwise, changes in this route set from iteration to iteration cause the flow patterns to fluctuate.

Secondly, we need to interpolate the weights of links starting from node j at the arrival time $t + c_{(i,j)}(t)$ or $w_{(j,k)}[t + c_{(i,j)}(t)]$, because $t + c_{(i,j)}(t)$ does not necessarily correspond to a discrete time point. Note that, in the case of instantaneous travel costs, we do not need any such interpolation because the weights of links starting from a node j are calculated at time t rather than $t + c_{(i,j)}(t)$.

4.2. Diagonalisation method

It is practically difficult to evaluate the above variational inequality (17) directly. In this respect, Ran and Boyce (1996, p. 193) suggested a mathematical programme that is equivalent to the variational inequality (17) under the diagonalisation method. This mathematical programme in discrete time can be written as follows: Note that the objective function (23a)–(23c) is constructed in terms of route flows and costs rather than link ones:

$$\min_{\mathbf{f}} \quad z(\mathbf{f}) = \sum_t \sum_{od} \left\{ -q^{od}(t) S^{od}\{\mathbf{C}^{od}(t)\} + \sum_p f_p^{od}(t) C_p^{od}(f_p^{od}(t)) - \sum_p \int_o^{f_p^{od}(t)} C_p^{od}(w)\,\mathrm{d}w \right\} \mathrm{d}t \tag{23a}$$

$$\text{subject to} \quad \sum_{p \in R_{od}} f_p^{od}(t) = q^{od}(t) \quad \forall od,\ \forall t \tag{23b}$$

$$f_p^{od}(t) \geqslant 0 \quad \forall od,\ \forall t \tag{23c}$$

where, $S^{od}\{\mathbf{C}^{od}(t)\}$ represents the expected perceived travel cost (dissatisfaction).

In the case of the logit model, Williams (1977) showed that it can be calculated as:

$$S^{od}\{\mathbf{C}^{od}(t)\} = -\frac{1}{\theta}\ln\sum_{p\in R_{od}}\exp[-\theta C_p^{od}(t)] \tag{24}$$

The solution to the SDUE assignment problem can be achieved by solving the mathematical program (20) for the separable link cost case within the framework of the diagonalisation method. That means we fix the flows and correspondent travel costs for all time increments at the previous iteration when we calculate new flow pattern at the current iteration. The diagonalisation method for the solution to the SDUE assignment can be written as:

Step 0 (initialisation): Set $m = 0$; find a feasible link flow vector $\mathbf{e}^m(\mathbf{t})$ with free-flow travel costs.

Step 1 (subproblem): Solve the diagonalised problem in terms of time variable by an existing algorithm for the symmetric (or separable) case from the earliest time to the latest time. This yields a new link flow vector $\mathbf{e}^{m+1}(\mathbf{t})$.

Step 2 (convergence test): If $|\mathbf{e}^{m+1}(\mathbf{t}) - \mathbf{e}^m(\mathbf{t})|$ is less than a convergence criterion ε, stop. If not, set $m = m + 1$, and go to Step 1.

The subproblem of the diagonalisation method (Step 1) can be solved by the following descent feasible direction algorithm (Luenberger, 1984, p. 214; Sheffi, 1985, p. 323):

Step 1.0 (initialisation): Set inner iteration counter $n = 1$; set $\mathbf{e}^n(\mathbf{t}) = \mathbf{e}^m(\mathbf{t})$.

Step 1.1 (update): Calculate link travel costs, $\mathbf{c}^n(\mathbf{t})$, using $\mathbf{e}^n(\mathbf{t})$.

Step 1.2 (direction finding): Implement a stochastic dynamic network loading based on the $\mathbf{c}^n(\mathbf{t})$, this yields auxiliary route flow $\hat{\mathbf{f}}^n(\mathbf{t})$ and auxiliary link flow $\mathbf{y}^n(\mathbf{t})$.

Step 1.3 (line search): Find λ^n as the value of λ that solves the following minimisation problem (25) in the increasing order of t, while fixing $\mathbf{f}^n(s)$, $\mathbf{C}^n(s)$ for $s \neq t$ at the current iteration n.

$$\arg\min_{0\leqslant\lambda\leqslant 1} z\left[\mathbf{f}^n(t) + \lambda\left\{\hat{\mathbf{f}}^n(t) - \mathbf{f}^n(t)\right\}\right] \quad \forall t \tag{25}$$

Then, update the route flow as

$$\mathbf{f}^{n+1}(t) = \mathbf{f}^n(t) + \lambda^n(t)\left(\hat{\mathbf{f}}^n(t) - \mathbf{f}^n(t)\right) \quad \forall t \tag{26}$$

Find a correspondent link flow pattern $\mathbf{e}^n(\mathbf{t})$ after a stochastic dynamic network loading with $\mathbf{f}^{n+1}(\mathbf{t})$.

Step 1.4 (convergence criteria): If n has reached a pre-specified number or satisfies the convergence criteria, stop; otherwise, set $n = n + 1$ and go to Step 1.1.

Because we perform stochastic dynamic network loading rather than the all-or-nothing assignment to find the search direction in Step 1.2, the difference between the current and the auxiliary flow pattern tends to zero over the iterations (Maher and Hughes, 1997). In other words, the auxiliary flow pattern from the stochastic dynamic network loading will be identical to the current flow pattern in an equilibrium state. Therefore, the difference between the current and the auxiliary flow pattern can be used for the convergence criteria (ρ_s) as:

$$\rho_s = \sum_t \sum_a \left[\frac{e_a^n(t) - y_a^n(t)}{(e_a^n(t) + y_a^n(t))}\right]^2 \Delta t \leqslant \varepsilon \tag{27}$$

Here we investigate difficulties of solving the above subproblem (Step 1) and suggest how to deal with them in two respects.

Firstly, it seems that we have to store route flow patterns $\mathbf{f}^n(\mathbf{t})$ and $\hat{\mathbf{f}}^n(\mathbf{t})$ in order to update a flow pattern in Step 1.3. However, we do not have to deal with route flows directly if we exploit DYNASTOCH algorithm. The main advantage of this algorithm is that we can deal with route choice probabilities or route flows implicitly without direct path enumeration. In this algorithm, we just calculate link choice probabilities at each node, and divide traffic between links at each node accordingly. These link choice probabilities correspond to the choice probabilities for reasonable routes, which were defined in advance. Therefore, if we combine link choice probabilities rather than link flows according to a move size λ, and perform forward pass according to those probabilities from the origin to the destination and from the initial instant to the final instant, we can update a link flow pattern without knowing route flows, while maintaining correct flow propagation. The way of updating link flows, which is correspondent to the updated route flows can be explained as below.

According to the DYNASTOCH algorithm, the link choice probability for link (i,j) at time t can be represented with link weights as:

$$p_{(i,j)}(t) = \frac{w_{(i,j)}(t)}{\sum_{(i,k)\in A(i)} w_{(i,k)}(t)} \tag{28}$$

Therefore, we can update link flows in the forward pass using updated link choice probabilities, $p^{n+1}_{(i,j)}(t)$ as:

$$e^{n+1}_{(i,j)}(t) = \begin{cases} q^{od}(t)p^{n+1}_{(i,j)}(t) & \text{if } i = o \\ \sum_{(k,i)\in B(i)} g^n_{(k,i)}(t)p^{n+1}_{(i,j)}(t) & \text{otherwise} \end{cases} \tag{29}$$

where,

$$p^{n+1}_{(i,j)}(t) = \frac{w^{n+1}_{(i,j)}(t)}{\sum_{(i,k)\in A(i)} w^{n+1}_{(i,k)}(t)} = \lambda^n \frac{u^n_{(i,j)}(t)}{\sum_{(i,k)\in A(i)} u^n_{(i,k)}(t)} + (1-\lambda^n)\frac{w^n_{(i,j)}(t)}{\sum_{(i,k)\in A(i)} w^n_{(i,k)}(t)} \quad \forall (i,j) \tag{30}$$

where, $u^n_{(i,j)}(t)$ denotes link weight for (i,j) at time t at iteration n corresponding to the auxiliary flow pattern.

Although we can calculate updated link choice probability $p^{n+1}_{(i,j)}(t)$ according to (30), we cannot directly obtain the link weight $w^{n+1}_{(i,j)}(t)$ which corresponds to $p^{n+1}_{(i,j)}(t)$. This link weight should be calculated in the current iteration so as to update flows (Step 1.3) in the next iteration. In this study, $w^{n+1}_{(i,j)}(t)$ is calculated approximately as:

$$w^{n+1}_{i,j}(t) = p^{n+1}_{(i,j)}(t)W^{n+1}_j(t) \tag{31}$$

where,

$$W^{n+1}_j(t) = \sum_{(j,k)\in A(j)} w^{n+1}_{(j,k)}(t) \cong \lambda \sum_{(j,k)\in A(j)} u^n_{(j,k)}(t) + (1-\lambda)\sum_{(j,k)\in A(j)} w^n_{(j,k)}(t) \tag{32}$$

Secondly, we note that it would be inefficient to apply interval reduction methods such as the golden section method or bi-section method (Sheffi, 1985) to find a move size λ in Step 1.3 or (25).

This is because we need to evaluate the objective function in terms of route quantities whenever λ value changes. However, we can decide move size λ without evaluation of the objective function in the stochastic case if we decide a move size according either to an interpolation method (Maher and Hughes, 1997) or to the method of successive average (MSA, Sheffi, 1985). Additionally, we can consider the pure stochastic dynamic network loading method, in which we fix the move size as 1 regardless of the number of the iteration. This means that we perform the DYNASTOCH algorithm successively without considering the move size in each iteration. We now discuss in turn these methods which can be used for the decision rule of the move size (23a)–(23c).

4.3. Quadratic interpolation method

We can calculate an appropriate move size using an interpolation method based upon either a quadratic or a cubic approximation to the objective function at time t with respect to λ. According to the quadratic interpolation method (Maher and Hughes, 1997), the appropriate move size $\hat{\lambda}(t)$ at time t, can be obtained from:

$$\hat{\lambda}(t) = \frac{-g_0(t)}{-g_0(t) + g_1(t)} \tag{33}$$

where, $g_0(t)$ is the gradient of $z(\mathbf{f}(t))$ at $\lambda = 0$, and $g_1(t)$ is the gradient of $z(\mathbf{f}(t))$ at $\lambda = 1$.

Generally, $g_\lambda^n(t)$, gradient of $z(\mathbf{f}(t))$ at iteration n with respect to λ can be calculated by:

$$\begin{aligned} g_\lambda^n(t) &= \frac{\mathrm{d}}{\mathrm{d}\lambda} z[\mathbf{f}^n(t) + \lambda(\hat{\mathbf{f}}^n(t) - \mathbf{f}^n(t))] \\ &= -\sum_{od}\sum_{p} (\hat{f}_p^\lambda(f_p^\lambda(t)) - f_p^\lambda(t)) \left.\frac{\mathrm{d}C_p}{\mathrm{d}f_p}\right|_{f_p = f_p^\lambda(t)} (\hat{f}_p(t) - f_p(t)) \end{aligned} \tag{34}$$

where, $f_p(t)$ is the current route inflow at instant t at iteration n, $\hat{f}_p(t)$ is the auxiliary route inflow of $f_p(t)$, $f_p^\lambda(t)$ is the updated route inflow with respect to λ, and $\hat{f}_p^\lambda(t)$ is the auxiliary route inflow of $f_p^\lambda(t)$, and can be obtained from additional network loading.

According to Eq. (34), we need to store route quantities and evaluate a derivative of route costs with respect to route inflows in order to calculate a gradient $g_\lambda^n(t)$. However, this equation can be written in terms of correspondent link quantities without loss of consistency if we assume that link costs are separable and increasing with respect to flows under diagonalisation method. This can be written as follows:

$$g_\lambda^n(t) \approx -\sum_a (\hat{y}_a(e_a^\lambda(t)) - e^\lambda(t)) \left.\frac{\mathrm{d}c_a^n}{\mathrm{d}e_a^n}\right|_{e_a^n = e_a^\lambda(t)} (y_a^n(t) - e_a^n(t)) \tag{35}$$

where, $e_a^n(t)$ is the current inflow at instant t at iteration n, and it is correspondent to $f_p(t)$, $y_a^n(t)$ is an auxiliary inflow of $e_a^n(t)$, and it is correspondent to $\hat{f}_p(t)$, $e_a^\lambda(t)$ is obtained from an additional dynamic network loading with updated link choice probabilities, which is decided by λ, and it is correspondent to $f_p^\lambda(t)$, $\hat{y}_a(e_a^\lambda(t))$ is the auxiliary flows from $e_a^\lambda(t)$, and obtained from performing additional dynamic network loading based on the cost pattern $\hat{\mathbf{c}}(\mathbf{t})$ resulting from $\mathbf{e}^\lambda(\mathbf{t})$, and it is correspondent to $\hat{f}_p^\lambda(t)$.

Therefore, $g_0^n(t)$ and $g_1^n(t)$ are obtained as:

$$g_0^n(t) = -\sum_a (y_a^n(t) - e_a^n(t))^2 \left.\frac{\mathrm{d}c_a^n}{\mathrm{d}e_a^n}\right|_{e_a^n = e_a^n(t)} \tag{36}$$

$$g_1^n(t) = -\sum_a (\hat{y}_a(y_a^n) - y_a^n(t)) \left.\frac{\mathrm{d}c_a^n}{\mathrm{d}e_a^n}\right|_{e_a^n = y_a^n(t)} (y_a^n(t) - e_a^n(t)) \tag{37}$$

Note that the quadratic interpolation (30) cannot be used for the reactive cost-flow association because the costs are not sensitive to the flows associated with them so that gradients $g_0^n(t)$ and $g_1^n(t)$ would be zero.

4.4. Method of successive average

According to the MSA method, the move size at time t at iteration n is specified as:

$$\lambda^n(t) = \frac{1}{n} \tag{38}$$

Compared to the quadratic interpolation (33), the MSA does not require any additional network loading, and the move size is decided irrespective of the objective function form. Therefore, we can apply the MSA method for both predictive and reactive assignments.

4.5. Pure network loading

In the case of pure network loading method, we fix $\lambda^n = 1.0$ regardless of the iteration number n. That means we do not update the link flow pattern as the one which lies between the current and the auxiliary flow pattern in each iteration. Instead, we just replace the current flow pattern with the auxiliary one; therefore, we perform stochastic dynamic network loading successively without a detailed consideration of the move size λ. This method can be applied for both predictive and reactive assignment because we do not consider any mathematical formulation to decide move size.

5. Numerical example

In this section, we apply the SDUE assignment model which adopts the deterministic queuing model for a link performance function and the ideal travel cost concept for calculating a route travel cost. The solution of the SDUE assignment model is found by the diagonalisation method, and we just perform one iteration to solve the subproblem according to Sheffi's (1985) streamlined version of the diagonalisation method. For a stochastic dynamic network loading, we use the DYNASTOCH algorithm in order to perform a logit-based stochastic network loading efficiently.

5.1. Demand profile

Fig. 4 shows the demand profile, which increases at a constant rate w until time 10 (min), and maintains a peak of $10w$ (vehicles/min) until time 15 (min), then decreases at a constant rate to

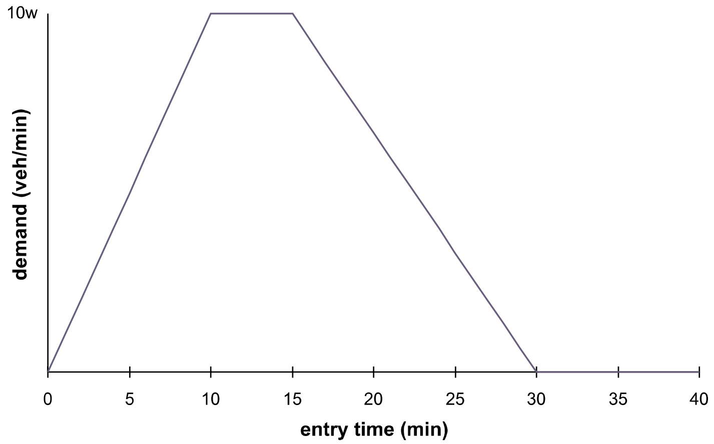

Fig. 4. Demand profile.

zero, which it reaches at time 30 (min). Note that we can change the size of demand conveniently by changing the value of w.

5.2. Two-link network

In order to apply the SDUE assignment model to the two-link network which has only one origin–destination pair we use the demand profile shown in Fig. 4, and we take the value of $w = 5.0$. Table 1 shows the specification of routes (or links) in the two-link network.

In the application of the SDUE assignment to the two-link network, we compare several ways of deciding move size λ in the line search (25): these are the quadratic interpolation, the MSA, and the pure network loading method. At the same time, we examine how the resulting flow patterns from predictive and reactive cost-flow association are different from each other.

First of all, the convergence of both predictive and reactive assignments associated with the various methods of deciding move size λ are shown in Tables 2–6 for various values of the dispersion parameter θ and various sizes of time increment Δt. Each cell in those tables represents the number of iterations required to achieve convergence and the difference between the current and auxiliary flow patterns at the final iterations (ρ_s), respectively. The maximum number of iterations and the convergence criteria (ε) are fixed at 50 and 0.0001, respectively.

We note that the convergent solutions cannot be found for all θ values or for all sizes of time increment. For example, Table 2 shows that we cannot find convergent solutions for $\theta \geqslant 2.5$ (min^{-1}) in the case of the quadratic interpolation when $\Delta t = 2.0$ (min) or for $\theta \geqslant 5.0$ (min^{-1}) when

Table 1
Specification of routes in two-link network

Route	Free-flow travel time, ϕ (min)	Capacity, Q (vehicles/min)
1	3	20
2	5	15

Table 2
Convergence of predictive assignment with quadratic interpolation (number of iterations, ρ_s)

Δt (min)	θ (min^{-1})				
	0.01	0.1	1.0	2.5	5.0
0.5	3, 0.000052	5, 0.000000	14, 0.000062	26, 0.000082	50, 0.003318
1.0	3, 0.000001	5, 0.000000	13, 0.000037	20, 0.000018	50, 1272.404
2.0	3, 0.000003	4, 0.000033	13, 0.000003	50, 3795.865	50, 3856.864

Table 3
Convergence of predictive assignment with MSA (number of iterations, ρ_s)

Δt (min)	θ (min^{-1})				
	0.01	0.1	1.0	2.5	5.0
0.5	45, 0.000097	35, 0.000094	50, 0.333966	50, 13.09109	50, 68.08843
1.0	30, 0.000100	23, 0.000093	50, 0.105733	50, 3.901663	50, 27.84309
2.0	48, 0.000098	43, 0.000095	50, 0.140691	50, 3.078960	50, 36.98283

Table 4
Convergence of reactive assignment with MSA (number of iterations, ρ_s)

Δt (min)	θ (min^{-1})				
	0.01	0.1	1.0	2.5	5.0
0.5	46, 0.000097	41, 0.000097	50, 0.913152	50, 51.55889	50, 1005.814
1.0	31, 0.000099	31, 0.000100	50, 1.663520	50, 254.7121	50, 630.9620
2.0	50, 0.000108	50, 0.000158	50, 156.3957	50, 1679.695	50, 2112.663

Table 5
Convergence of predictive assignment with pure network loading (number of iterations, ρ_s)

Δt (min)	θ (min^{-1})				
	0.01	0.1	1.0	2.5	5.0
0.5	4, 0.000015	10, 0.000015	50, 3614.769	50, 4660.925	50, 4922.582
1.0	4, 0.000021	11, 0.000011	50, 5081.883	50, 5684.758	50, 5777.283
2.0	4, 0.000037	12, 0.000070	50, 5837.981	50, 6144.085	50, 6239.511

Table 6
Convergence of reactive assignment with pure network loading (number of iterations, ρ_s)

Δt (min)	θ (min^{-1})				
	0.01	0.1	1.0	2.5	5.0
0.5	4, 0.000010	9, 0.000007	35, 0.000003	47, 0.000005	46, 0.000011
1.0	4, 0.000009	8, 0.000052	24, 0.000035	25, 0.000007	23, 0.000000
2.0	4, 0.000007	7, 0.000099	13, 0.000000	12, 0.000000	11, 0.000000

$\Delta t = 1.0$ (min). Tables 3–5 show that we cannot find convergent solutions for $\theta \geqslant 1.0$ (min^{-1}) in the case of the predictive MSA, the reactive MSA, and the predictive pure network loading method respectively. In contrast, Table 6 shows that we can find convergent solutions in all cases by using the reactive pure network loading method. These results show that not all method for deciding move size can give rise to a convergent solution at the same θ value.

To examine how the resulting flow patterns differ from each other between predictive and reactive cost-flow association (or assignments), we plot the inflow patterns for link 1 against time for various values of θ. We can see from Figs. 5 and 6 that there is almost no difference between the convergent flow patterns from predictive and reactive assignments when the value of θ is small regardless of the method for deciding move size λ. However, as the value of θ increases, we can find substantial differences between predictive and reactive assignments from $\theta = 1.0$ (min^{-1}). For example, Fig. 7 shows the difference in the resulting flow patterns between predictive and reactive assignments, and this difference can be found irrespective of the solution method applied. Furthermore, from Figs. 8 and 9 we can see oscillating flow patterns for $\theta \geqslant 2.5$ (min^{-1}), in the case of

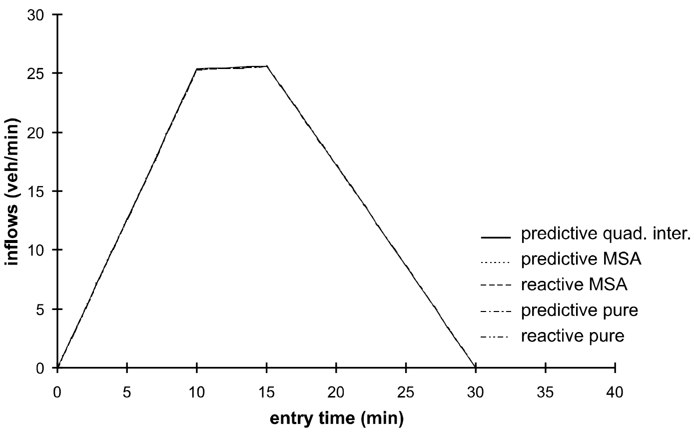

Fig. 5. Inflows to link 1 when $\theta = 0.01$, $\Delta t = 1.0$.

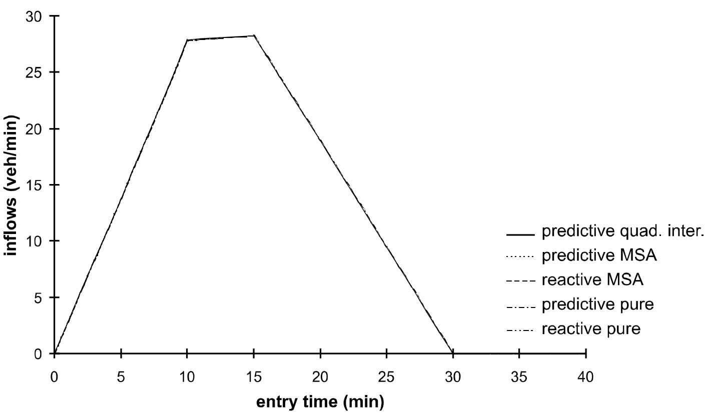

Fig. 6. Inflows to link 1 when $\theta = 0.1$, $\Delta t = 1.0$.

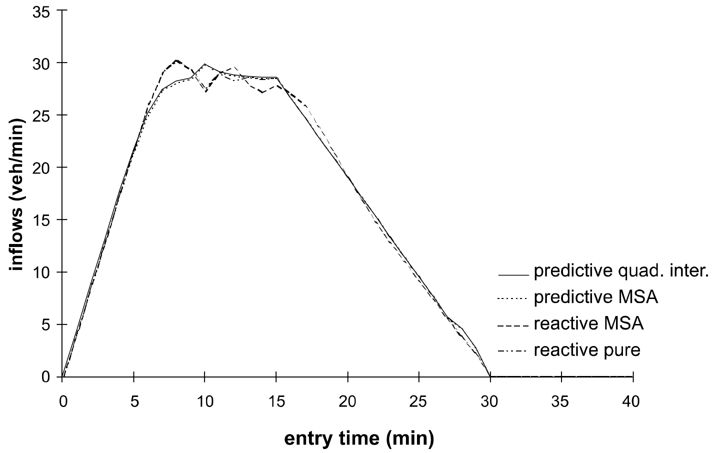

Fig. 7. Inflows to link 1 when $\theta = 1.0$, $\Delta t = 1.0$.

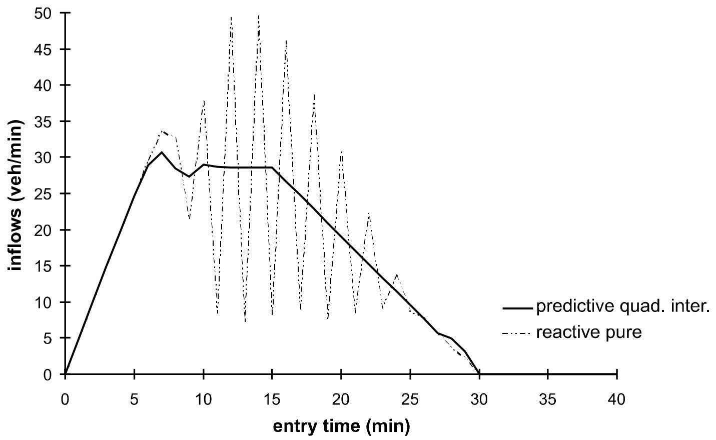

Fig. 8. Inflows to link 1 when $\theta = 2.5$, $\Delta t = 1.0$.

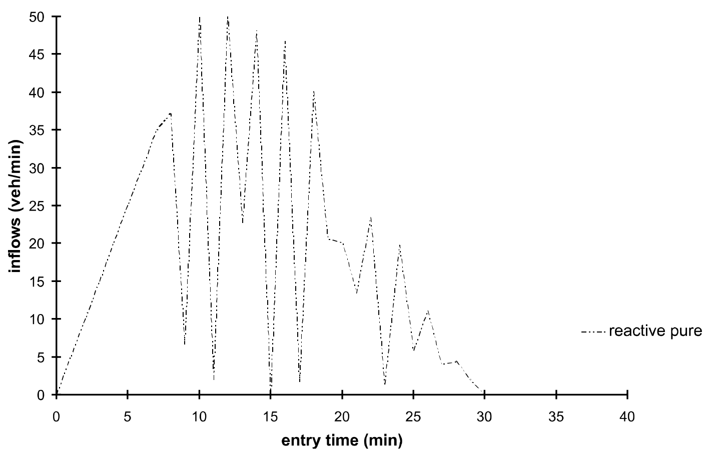

Fig. 9. Inflows to link 1 when $\theta = 5.0$, $\Delta t = 1.0$.

the reactive pure network loading method, unlike the predictive assignment. This means that although the reactive pure network loading method gives rise to convergent solutions even at large θ value unlike the other approaches, the resulting flow pattern is implausible.

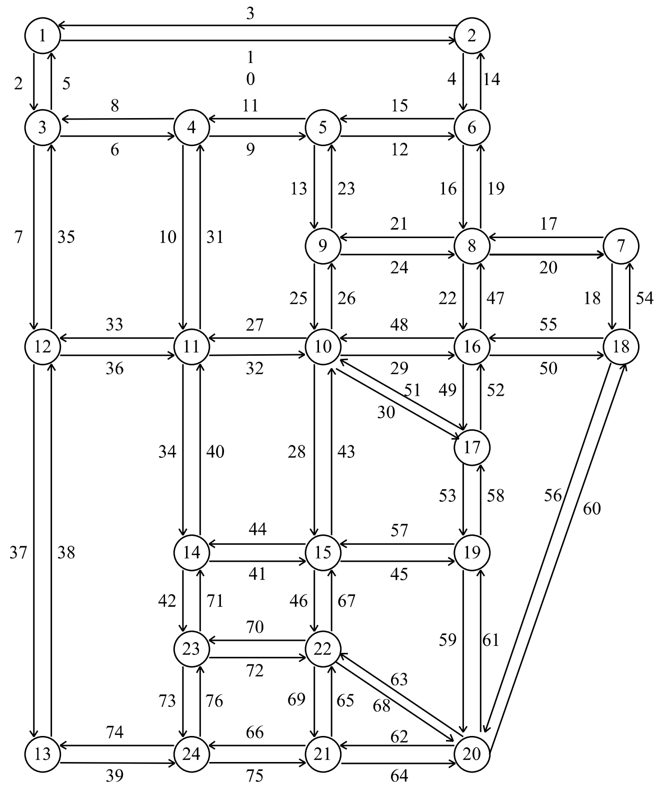

Fig. 10. Sioux Falls network.

Table 7
New origin–destination pairs for Sioux Falls network

Origin	Destination
1	10
4	19
6	15
7	15
12	19
13	10
14	8
18	5
20	9
22	8
2	15
3	16

5.3. Sioux Falls network

Fig. 10 shows the Sioux Falls network (LeBlanc, 1975) which has 24 nodes and 76 links. In this network, we just consider 12 origin–destination pairs specified in Table 7 and we use the same demand profile for all origin–destination pairs with the value of $w = 3.0$ in Fig. 4. The

Table 8
Specification of the Sioux Falls network

Link	A_node	B_node	Free-flow travel time (min)	Capacity (vehicles/min)	Link	A_node	B_node	Free-flow travel time (min)	Capacity (vehicles/min)
1	1	2	6	65	39	13	24	2	60
2	1	3	2	55	40	14	11	4	50
3	2	1	6	65	41	14	15	4	50
4	2	6	2	60	42	14	23	3	40
5	3	1	2	55	43	15	10	4	45
6	3	4	5	60	44	15	14	4	50
7	3	12	5	60	45	15	19	3	40
8	4	3	5	60	46	15	22	3	45
9	4	5	3	50	47	16	8	2	45
10	4	11	5	55	48	16	10	3	40
11	5	4	3	50	49	16	17	2	45
12	5	6	3	50	50	16	18	3	55
13	5	9	2	50	51	17	10	3	45
14	6	2	2	60	52	17	16	2	45
15	6	5	3	50	53	17	19	3	45
16	6	8	3	45	54	18	7	5	50
17	7	8	3	40	55	18	16	3	55
18	7	18	5	50	56	18	20	6	55
19	8	6	3	45	57	19	15	3	40
20	8	7	3	40	58	19	17	3	45
21	8	9	3	45	59	19	20	4	50
22	8	16	2	45	60	20	18	6	55
23	9	5	2	50	61	20	19	4	50
24	9	8	3	45	62	20	21	3	40
25	9	10	2	45	63	20	22	4	45
26	10	9	2	45	64	21	20	3	40
27	10	11	5	50	65	21	22	2	50
28	10	15	4	45	66	21	24	3	50
29	10	16	3	40	67	22	15	3	45
30	10	17	3	45	68	22	20	4	45
31	11	4	5	55	9	22	21	2	50
32	11	10	5	50	70	22	23	4	40
33	11	12	3	60	71	23	14	3	40
34	11	14	4	50	72	23	22	4	40
35	12	3	5	60	73	23	24	2	40
36	12	11	3	60	74	24	13	2	60
37	12	13	6	65	75	24	21	3	50
38	13	12	6	65	76	24	23	2	40

Table 9
Convergence of SDUE assignment over various θ value in Sioux Falls network

	θ (min^{-1})		
	0.01	0.04	0.1
Number of iterations (ρ_s)	3, 0.000015	6, 0.000016	50, 1.670002

specification of the Sioux Falls network is summarised in Table 8. Note that this specification is newly designed based on the original specification in LeBlanc (1975).

In the application of the SDUE assignment to the Sioux Falls network, we consider only the predictive rather than the reactive assignment because the former shows stable solution irre-

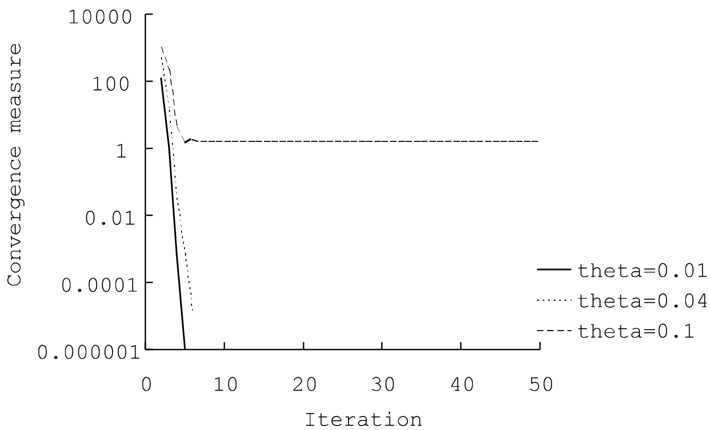

Fig. 11. The value of convergence measure (ρ_s) over iterations in Sioux Falls network.

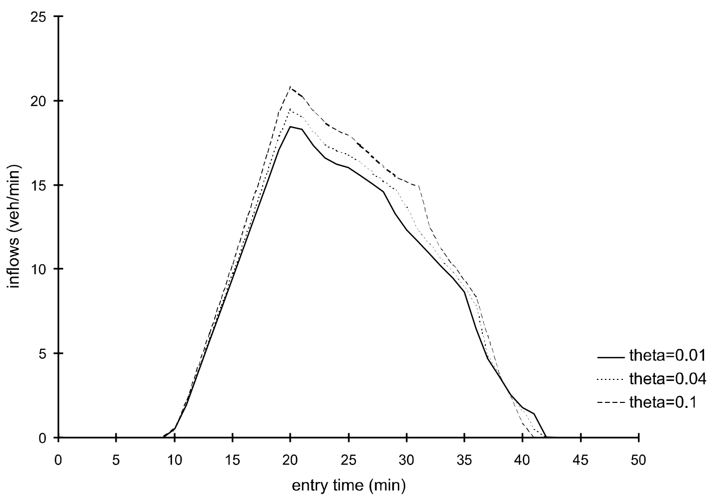

Fig. 12. Inflows to link 24 for various values of θ.

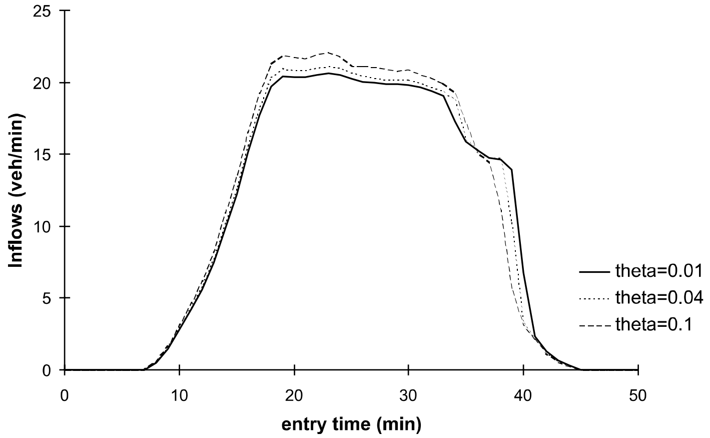

Fig. 13. Inflows to link 29 for various values of θ.

spective of the value of dispersion parameter θ once it gives rise to convergent solution. In particular, only the quadratic interpolation is considered for the method for deciding move size λ, because it gives convergent solution even in the relatively large values of θ compared to the other methods.

Table 9 and Fig. 11 show the convergence of this SDUE assignment for various values of θ. We cannot find a solution that meets the convergence criterion $\varepsilon = 0.0001$ when $\theta = 0.1$ (min^{-1}), but, we use the resulting flow pattern when $\theta = 0.1$ (min^{-1}) for the sake of comparison with those from other values of θ. This seems to be acceptable in that the value of ρ_s is not so large. Figs. 12 and 13 show how the flow pattern varies as the value of θ for link 24 and 29 respectively. In both cases, we can find smooth flow patterns for all θ values and increasing flow pattern as the value of θ increases. In terms of the computation times, it has taken only 12 min to calculate convergent solution of the SDUE assignment when we code the model in C/C++ and run it on the Pentium III personal computer with 450 MHz CPU speed and 64 MB RAM under the Microsoft Windows 3.1 operating system.

6. Conclusion

In this study, we have seen what is required in dynamic traffic modelling and proposed the deterministic queuing model for a plausible link performance function because this model can maintain causality, the FIFO discipline, and correct flow propagation. Then, the SDUE assignment problem is solved without direct evaluation of the objective function in the framework of the diagonalisation method. For example, we can adopt a quadratic interpolation method which decides move size with single additional network loading in the solution algorithm. For a stochastic dynamic network loading, this study developed the DYNASTOCH algorithm (Ran and Boyce, 1996) so that it can be applied to a general network while maintaining correct flow propagation. Furthermore, it was examined how we can update a link flow pattern rather than a route one in the solution algorithm using the DYNASTOCH algorithm.

Application of the developed SDUE assignment model to a two-link network showed that the predictive cost-flow association is preferable to the reactive one in that the former case gives rise to a convergent flow pattern which is smooth irrespective of the size of the dispersion parameter θ in logit model, but the latter case gives rise to a convergent flow pattern which oscillates as the value of θ increases. The application to the Sioux Falls network showed that the present SDUE assignment model could be used for general networks with plausible results at the reasonable computation time.

For future studies, it seems to be desirable to incorporate other types of link performance function, to apply the present SDUE assignment model for other realistic networks and to evaluate the effect of various traffic management measures such as ramp metering and variable message signs which vary according to the prevailing network conditions.

Acknowledgements

The author is very grateful to Professor B.G. Heydecker for his encouragement and supervision on the research, and to Professor R.E. Allsop for his helpful discussions.

References

Astarita, V., 1996. A continuous time link model for dynamic network loading based on travel time functions. In: Lesort, J.B. (Ed.), Transportation and Traffic Theory. Pergamon, Oxford, pp. 79–102.

Ben-Akiva, M., De Palma, A., Kanaroglou, P., 1986. Dynamic model of peak period traffic congestion with elastic arrival rates. Transportation Science 20 (2), 164–181.

Buisson, C., Lebacque, J.P., Lesort, J.B., 1998. Travel times computation for dynamic assignment modelling. In: Bell, M.G.H. (Ed.), Transportation Networks: Recent Methodological Advances. Pergamon, Oxford, pp. 303–317.

Cascetta, E., Cantarella, G.E., 1991. A day-to-day and within-day dynamic stochastic assignment model. Transportation Research 25A (5), 277–291.

Daganzo, C.F., Sheffi, Y., 1977. On stochastic models of traffic assignment. Transportation Science 11 (3), 253–274.

De Palma, A., Ben-Akiva, M., Lefevre, C., Litinas, N., 1983. Stochastic equilibrium model of peak period traffic congestion. Transportation Science 17 (4), 430–453.

Dial, R.B., 1971. A probabilistic multipath traffic assignment model which obviates path enumeration. Transportation Research 5 (2), 83–111.

Friesz, T.L., Bernstein, D., Smith, T.E., Tobin, R.L., Wie, B., 1993. A variational inequality formulation of the dynamic networks user equilibrium problem. Operation Research 41 (1), 179–191.

Han, S., 2000. Dynamic traffic assignment techniques for general road networks. Ph.D. Thesis, University of London, UK, submitted for examination in August, 2000.

Heydecker, B.G., Addison, J.D., 1996. An exact expression of dynamic traffic equilibrium. In: Lesort, J.B. (Ed.), Transportation and Traffic Theory. Pergamon, Oxford, pp. 359–383.

Heydecker, B.G., Addison, J.D., 1998. Analysis of traffic models for dynamic equilibrium traffic assignment. In: Bell, M.G.H. (Ed.), Transportation Networks: Recent Methodological Advances. Pergamon, Oxford.

Heydecker, B.G., Verlander, N., 1999. Calculation of dynamic traffic equilibrium assignments. Proceedings of the European Transport Conferences, Seminar F, P434, pp. 79–91.

Kuwahara, M., Akamatsu, T., 1997. Decomposition of the reactive dynamic assignment with queues for a many-to-many origin–destination pattern. Transportation Research 31B (1), 1–10.

LeBlanc, L.J., 1975. An algorithm for the discrete network design problem. Transportation Science 9 (3), 183–199.

Luenberger, D.G., 1984. Linear and nonlinear programming. Addison-Wesley, Reading, Massachusetts.

Maher, M.J., Hughes, P.C., 1997. A probit-based stochastic user equilibrium assignment model. Transportation Research 31B (4), 341–345.

Nagurney, A., 1993. Network Economics: A Variational Inequality approach. Kluwer Academic Publishers, Norwell, MA.

Newell, G.F., 1982. Application of queuing theory. Chapman and Hall, London.

Ran, B., Boyce, D.E., 1996. Modelling Dynamic Transportation Networks: An Intelligent Transportation System Oriented Approach. Springer, Berlin.

Sheffi, Y., 1985. Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods. Prentice-Hall, Englewood Cliffs, New Jersey.

Sheffi, Y., Powell, W., 1982. An algorithm for the equilibrium assignment problem with random link times. Networks 12 (2), 191–207.

Vythoulkas, P., 1990. A dynamic stochastic assignment model for the analysis of general networks. Transportation Research 24B (6), 453–469.

Williams, H.C.W.L., 1977. On the formation of travel demand models and economic evaluation measures of user benefit. Environment and Planning 9A, 285–344.