한상진 ● 교통개발연구원 도로교통연구실 책임연구원

경로기반 해법알고리즘을 이용한 동적통행배분모형의 개발

대한교통학회
제41회

A ROUTE-BASED SOLUTION ALGORITHM FOR DYNAMIC USER EQUILIBRIUM ASSIGNMENT

Sangjin Han[1]

Centre for Transport Studies

UCL (University College London) Gower Street, London WC1E 6BT

United Kingdom

ABSTRACT

The aim of the present study is to find a good quality user equilibrium assignments under time varying condition. For this purpose, this study introduces a dynamic network loading method that can maintain correct flow propagation as well as flow conservation, and it develops a novel solution algorithm that does not need evaluation of the objective function by modifying the Schittenhelm (1990)'s algorithm. This novel algorithm turns out to be efficient and convenient compared to the conventional Frank-Wolfe (1956) algorithm because the former finds solutions based on routes rather than links so that it can maintain correct flow propagation intrinsically in the time-varying network conditions. The application of dynamic user equilibrium (DUE) assignment model with this novel solution algorithm to test networks including medium-sized one shows that the present DUE assignment model gives rise to high quality discrete time solutions when we adopt the deterministic queuing model for a link performance function, and we associate flows and costs in a proper way.

Keywords: Dynamic deterministic user equilibrium assignment; Solution algorithm; Flow propagation; Deterministic queuing model; Link performance function; Discrete time

---

[1] Present address: Korea Transport Institute, 2311 Daehwadong, Ilsangu, Koyangsi 411-701, Korea.

Tel:+82-31-910-3112; Fax: +82-31-910-3235; E-mail: han@koti.re.kr

1. INTRODUCTION

Dynamic traffic assignment is useful tool to analyse how congestion forms and dissipates in time-varying conditions. In particular, this is necessary for the evaluation of various ITS (Intelligent Transport System) measures which usually require taking into account congestion and other variations over time.

Numerous mathematical formulations have been proposed to define the dynamic user equilibrium state. In general, they can be grouped into three categories; mathematical programmes, optimal control theory, and variational inequality. For example, Merchant and Nemhauser (1978), Janson (1991), Kuwahara and Akamatsu (1993), Jayakrishnan et al (1995), and Lin and Lo (2000) formulated the dynamic assignment problem as a mathematical programme. On the other hand, Friesz et al (1989), Wie et al (1990), Papageorgiou (1990), Wie et al (1994), and Lam and Huang (1995) formulated the same problem as an optimal control problem. Finally, Smith (1993), Friesz et al (1993), Ran and Boyce (1996), Chen and Hsueh (1998), and Heydecker and Verlander (1999) adopted variational inequality (VI).

In order to solve the dynamic assignment problem, especially when the formulation is based on a mathematical programme or a VI, many studies adopted the Frank-Wolfe (F-W) (1956) algorithm. For example, Janson (1991), Ran and Boyce (1996a, b), Jayakrishnan et al (1995), and Chen and Hsueh (1998) all applied the F-W algorithm in their solution approach. However, it has been known that this algorithm gives rise to very slow convergence after a few iterations because of the zigzagging pattern of the descent direction (Patriksson, 1994).

There have been mainly two approaches as an alternative to the F-W algorithm. One is the column generation method (Dafermos and Sparrow, 1969; Leventhal et al, 1973), the other is the more general Simplicial Decomposition (SD) method (Larsson and Patriksson, 1992; Patriksson, 1994; Lee, 1995).

The main idea of these methods is that a progress towards the solution can be made more efficiently by using several extreme points rather than a single one as in the F-W algorithm. It has been known that both solution approaches usually give rise to more accurate equilibrium solution at less computation time compare to the F-W algorithm.

In particular, when it comes to the dynamic assignment, a column generation method that finds a solution based on route flows rather than link ones, becomes more efficient compared to the link-based solution approaches such as the F-W algorithm and SD method. This is because the route-based solution approach maintains correct flow propagation intrinsically unlike the link-based one. In this respect, the present study develops a novel route-based solution algorithm by modifying Schittenhelm's (1990) algorithm, a kind of the column generation method. The main idea of this novel solution algorithm is to use an interpolation method to find the optimal move size. By doing so, we can avoid repeated objective function evaluation in the line-search step. Therefore, the novel algorithm is particularly advantageous when it is used to solve a dynamic assignment problem where the evaluation of an objective function requires more computation time compared to its static counterpart. Furthermore, this study introduces and explains in detail a dynamic network loading method that maintains correct flow propagation as well as flow conservation condition.

In the present dynamic assignment model, the deterministic queuing model is used as the link performance function and the 'ideal' rather than the 'instantaneous' one is used in order to reflect conditions that travellers encounter during their journeys (Ran and Boyce, 1996). Furthermore, the present analysis shows how to associate costs with flows in each discrete time increment.

The structure of the present paper is as follows. In Section 2, the formulation of the dynamic assignment problem is presented, and Section 3 explains a dynamic network loading method in detail with a flow chart. Then Section 4 introduces a novel route-based solution algorithm, and compares it with a link-based solution approach. The results of the application to simple test networks are shown in Section 5.

2. FORMULATION OF THE MODEL

We can define the dynamic user equilibrium (DUE) state according to the Wardrop's (1956) principle as follows :

"The travel costs incurred by traffic on all routes entered by traffic at each instant are equal or less than those that would be on any unused route at that instant."

We can write the above statement in a mathematical form after Beckmann (1956) as:

$$f_p(t)\begin{cases} >0 & \Rightarrow C_p(t) = C^*_{od}(t) \\ =0 & \Rightarrow C_p(t) \geq C^*_{od}(t) \end{cases} \quad \forall p \in R_{od} \quad \forall od \quad \forall t \tag{1}$$

$f_p(t)$ is an instantaneous flow entering route p at time t

R_{od} is the set of all routes from origin o to destination d

$C_p(t)$ is the cost incurred on route p from o to d starting at time t

$C^*_{od}(t)$ is the minimum travel cost from o to d starting at time t

The complementary inequality (1) states that in the DUE state, the flow entering a route p at time t would be greater than 0 only if the travel cost on the route is equal to the minimum travel cost at that time.

This complementary inequality (1) can be transformed into the variational inequality form according to Smith (1979). Thus an assignment at time t, expressed in the form of a column vector of route inflows $\mathbf{f}(t)$, is an equilibrium if and only if :

$$-[\mathbf{h} - \mathbf{f}(t)]^T \cdot \mathbf{C}(t) \leq 0 \tag{2}$$

for all feasible route flow vectors $\mathbf{h}$, where the feasibility condition for route flows $\mathbf{f}(t)$ are,

$$f_p(t) \geq 0 \qquad \forall\, p \in R_{od}$$

$$\sum_{p \in R_{od}} f_p(t) = q^{od}(t) \qquad \forall t \tag{3}$$

where, $q^{od}(t)$ is the demand for *od* at time *t*

We can transform the variational inequality (2) as a forward dynamic programme according to Heydecker and Verlander (1999). This can be explained as follows. First, we can see that for any $\mathbf{f}(t)$, the left-hand side of (2) will attain the value zero when $\mathbf{h} = \mathbf{f}(t)$. Therefore, if an assignment $\mathbf{f}(t)$ is found for which the greatest value in the left-hand side of (2) is zero, then $\mathbf{f}(t)$ satisfies (2) and so is an equilibrium. This means that the equilibrium solution, $\mathbf{f}^*(t)$ can be obtained by solving the equivalent minimisation problem as:

$$\min_{\mathbf{f} \in D(t)} \quad z(\mathbf{f}, t) \tag{4}$$

where, $D(t)$ is the set of route inflow vectors that satisfy the feasibility condition (3), and

$$z(\mathbf{f}, t) = \max_{\mathbf{h} \in D(t)} \quad -[\mathbf{h} - \mathbf{f}]^T \cdot \mathbf{C}(t) \tag{5}$$

Note that the objective function (5) will be exactly zero in equilibrium state according to the condition (2). We can also write the dynamic user equilibrium assignment problem over time in the form of a dynamic programme as:

$$\min_{\mathbf{f}} \int_t z(\mathbf{f}(t), t)\, dt \tag{6}$$

subject to

$$f(t) \in D(t) \qquad \forall t$$

Thus, the optimal value of the objective function $\int_t z(\mathbf{f}(t),t)dt$ is known to be zero and the value of $\mathbf{f}(t)$ that achieves this at time t can be calculated from $\mathbf{C}(t)$ using flows that entered at earlier times.

3. A DYNAMIC NETWORK LOADING METHOD

A network loading is a process which decides how flows will be distributed over the network with a given route flow pattern for all origin-destination pairs. In static assignment, this is simple because we assume that flows will be constant over the whole route regardless of time. However, dynamic network loading is not as straightforward as in the static case because we should maintain flow propagation as well as flow conservation. In other words, we should consider when and how much traffic from various routes in various origin-destination pairs at each instant will arrive at a certain node and how they will be split into connected links. Therefore, a key in dynamic network loading is how we collect traffic from various sources in a link and split them into connected links. In this study, a dynamic network loading is performed as described in Figure 1 with given routes and their flow patterns for each time increment.

Figure 1 A dynamic network loading method

For each time increment $[t, t+\Delta t)$
- For each origin-destination pair *od*
  - for each route *p*
    - for each successive link *a* within route *p*
      - $e_a(t) = e_a(t) + f_{p,a}^{od}(t)$
- For each link *a*
  - Update link cost $c_a(t)$
  - Calculate $h_{p,a}^{od}(\tau_a(t))$ with (7), (8) if $\delta_p^a = 1$, for $\forall\, p \in R_{od}$, $\forall\, od$;
    Set $f_{p,b}^{od}(\tau_a(t)) = h_{p,a}^{od}(\tau_a(t))$ *b* is the next link after *a* in route *p*

As can be seen in Figure 1, the link inflow $e_a(t)$ is calculated by summing all $f_{p,a}^{od}(t)$ which represent inflows to link a via route p for origin-destination pair od at instant t. In other words, we sum all possible route flows which enter link a at instant t for all origin-destination pairs to calculate link inflow $e_a(t)$.

Then we can obtain the exit time $\tau_a(t)$ by calculating travel cost $c_a(t)$ with inflows $e_a(t)$ and a link cost function. Once we know the exit time, we should calculate $g_a(\tau_a(t))$ the link outflow at the exit time according to the following flow propagation condition[2]:

$$g_a(\tau_a(t)) = \frac{e_a(t)}{\frac{d\tau_a(t)}{dt}} \tag{7}$$

Then we calculate all possible $h_{p,a}^{od}(\tau_a(t))$, which represent outflows from link a via route p for origin-destination pair od at exit time $\tau_a(t)$ using (7) and the following equation (8) which is proved by Kuwahara and Akamatus (1997):

$$\frac{h_p^{od}(\tau(t))}{g_a(\tau(t))} = \frac{f_p^{od}(t)}{e_a(t)} \qquad \text{for } \forall p \text{ which uses link } a \tag{8}$$

Finally, we set $h_{p,a}^{od}(\tau_a(t))$ equal to $f_{p,b}^{od}(\tau_a(t))$, where b is the next link after link a in the same route p. In this way, we can always calculate when and how much traffic will be split into the connected links from the downstream end of each link beforehand. That means we can calculate inflows to each link at each instant, maintaining correct flow propagation as well as flow conservation in this network loading by considering the time progressively from the earliest to the latest.

This dynamic network loading method needs path enumeration a priori. In this study, we propose to define the route set by the reasonable route in the sense of Dial (1971), and enumerate them using a 'back-tracking' algorithm (Steenbrink, 1974). This is explained in Appendix.

---

[2] The derivation of flow propagation condition (7) can be found in Han (2000a, b)

If we calculate the link travel cost $c_a(t)$ or exit time $\tau_a(t)$, it does not necessarily correspond to the discrete time (or integer number), but it is more likely to be a real number, because it is calculated from the link cost function, which allows real numbers. Therefore, we need to interpolate the value of $h^{od}_{p,a}(t)$ at each integer point from the values in the middle of the time increment, because practical solution procedures deal with discrete rather than continuous time. The method of interpolation is also explained in Appendix.

4. SOLUTION APPROACH

It is practically difficult to solve the continuous minimisation problem (6) analytically in general road networks. In this respect, this study converts the continuous minimisation problem (6) into a discrete one and solves it in the framework of triangularisation method. Firstly, the discrete version of the continuous minimisation problem (6) can be written as follows:

$$\min_{\mathbf{f}} \sum_t z(\mathbf{f}(t), t)$$

subject to (9)

$$f(t) \in D(t) \qquad \forall t$$

Where,

$$z(\mathbf{f},t) = \max_{\mathbf{h}\in D(t)} -[\mathbf{h}-\mathbf{f}]^T \cdot \mathbf{C}(t+\Delta t) \tag{10}$$

Note that flows at time t are associated with the costs at time $t+\Delta t$ rather than the costs at time t in the objective function (10). Heydecker and Verlander (1999) particularly called this cost-flow association a 'predictive' assignment in order to distinguish it from a 'reactive' assignment, which associates costs and flows at the same instant t. In other words, the difference between the predictive and the reactive assignment lies on the way of associating flows and costs in a discrete time increment $[t, t+\Delta t)$. The former associates flows during $[t, t+\Delta t)$ with the cost at the final instant $t+\Delta t$, whereas the latter associates flows during $[t, t+\Delta t)$ with the cost at the initial instant t. This cost-flow association becomes important when we deal with a time as a discrete variable rather than continuous one. Further detailed explanation on the matter of how to associate costs and flows in discrete time can be found Han (2000, 2002).

The optimal value of the objective function $z(\mathbf{f})$ in (10) would be exactly 0 at each instant in the equilibrium state according to variational inequality (2). Based on this property, we can establish that there is no contribution from future inflows to the summand in (9) at each time t. Furthermore, the assigned flows, $\mathbf{e}(t)$, influence the future costs but not past ones. That means we can find an optimal solution at time t without knowing the future optimal solutions after time t. Accordingly, the optimal solution of the dynamic programming (9) can be found by solving the individual step (4) separately in the increasing order of time (i.e. from the earliest time increment to the latest one).

The concept of this solution approach can be reflected in the diagonalisation method (Dafermos, 1982; Ran and Boyce, 1996), which is known to be a suitable method to solve the mathematical programme when the Jacobian matrix of the link cost functions is non-separable and asymmetric. However, we appeal to a triangularisation method akin to the Gauss-Seidel method rather than to a diagonalisation method, which would correspond to a Jacobi one. In the triangularisation method, the assignment calculated for each instant respect all earlier re-assignments at the present iteration rather than at the previous one.

The solution method based on triangularisation can be written as follows:

Step 0 (initialisation): Find a feasible link flow vector $\mathbf{e}^m(\mathbf{t})$ from free-flow travel costs, set m=0.

Step 1 (subproblem): Solve the triangularised problem with the existing algorithm for the separable case. This yields new link flow vector $\mathbf{e}^{m+1}(\mathbf{t})$.

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

In Step 1, any solution algorithms for the separable case can be applied. However, it should be noted that they should be applied sequentially from the earliest time instant to the latest one, and we should fix the flows until the processing time t as the ones which have been obtained at the current iteration rather than the flows in the previous iteration. We now consider two distinct ways in which the triangularisation problem in Step 1 can be solved. One is a link-based solution approach and the other is a route-based one.

4.1 A link-based approach (Frank-Wolfe algorithm)

The most popular link-based solution approach is the Frank-Wolfe (1956) (F-W) algorithm, and this can be written as follows.

Step 1.0 (initialisation of time increment): Set t=0.

Step 1.1 (initialisation of iteration counter): Set n=1.

Step 1.2 (update): Calculate the link travel costs, $\mathbf{c}^{n}(t+\Delta t)$,

using $\mathbf{e}^{n}(s)$ for $s \le t$, $\mathbf{e}^{n-1}(s)$ for $s > t$

Step 1.3 (direction finding): Implement 'all-or-nothing' assignment to find auxiliary link flow $\mathbf{y}^{n}(t)$, using fixed $\mathbf{c}^{n}(\mathbf{t}+\Delta t)$

Step 1.4 (line search): Find $\lambda^{n}(t)$ as the value of λ that solves

$$\min_{0\le\lambda\le1} \quad z[\mathbf{e}^{n}(t) + \lambda\{\mathbf{y}^{n}(t) - \mathbf{e}^{n}(t)\}]$$

Then, calculate new link inflow $\mathbf{e}^{n+1}(t)$ as,

$$\mathbf{e}^{n+1}(t) = \mathbf{e}^{n}(t) + \lambda^{n}(t)\{\mathbf{y}^{n}(t) - \mathbf{e}^{n}(t)\} \qquad \text{for all } t \tag{11}$$

Step 1.5 (convergence test for inner iteration): if n has reached a pre-specified number or satisfies the convergence criterion, set $t=t+1$ and go to Step 1.1; otherwise, set $n=n+1$ and go to Step 1.2

In this F-W algorithm, we update link flows by the linear combination of the current link flows and auxiliary link flows as equation (11). However, if we associate $\lambda(t)$ directly with link flows as (11), the updated link flows do not necessarily respect flow propagation because they do not result from a dynamic network loading. In fact, $\lambda(t)$ should be associated with route flows, and the corresponding link flows should be obtained from a network loading. This means we need to perform another network loading whenever $\lambda(t)$ changes in order to evaluate an objective function properly because it normally requires the updated link flows and associated costs. For example, a dynamic generalisation of Beckmann (1956)'s objective function, which was first shown in Janson (1991) requires updated link flows and associated costs. This objective function can be written as:

$$\sum_{t}\sum_{a}\int_{0}^{e_a(t)} c_a(w)dw \tag{12}$$

In this respect, it would be quite inefficient to adopt the F-W algorithm, and this problem can arise in any link-based solution approach for the dynamic assignment problem. Nevertheless, this problem has not so far been explicitly explained in the literature that has adopted the F-W algorithm as the solution method (e.g. Janson, 1991; Jayakrishnan et al, 1995; Ran and Boyce, 1996; Chen and Hsueh, 1998). Apart from these problems, it has been known that the F-W algorithm shows slow convergence near to the optimal solution because of the zigzagging pattern of the steepest descent direction (Patriksson, 1994).

4.2 A route-based approach (a novel solution algorithm)

Originally, route-based solution algorithms have been developed to overcome the slow convergence of the F-W algorithm. This algorithm is sometimes called a 'column generation method' in the sense that they find a solution using several route flow patterns (or columns), that have been generated over iterations. It has been known that the column generation method gives rise to an accurate equilibrium solution at a relatively small number of iterations compared to the F-W algorithm. This method was first applied to static assignment problems by Dafermos and Sparrow (1969) with explicit path enumeration. Then Leventhal et al (1973) developed the method which can generates routes, as they are needed. Later Schittenhelm (1990) applied this method to solve the combined trip distribution and assignment problem.

We can take advantage of a route-based solution approach particularly when we solve a dynamic assignment problem because it can maintain correct flow propagation intrinsicallly. This approach does not update link flows directly as we do in the F-W algorithm. But it updates route flows first, then it calculates link flows and associated costs after a dynamic network loading with updated route flows.

In this respect, this study develops a novel column generation method to solve a dynamic assignment problem by modifying the Schittenhelm's algorithm. Firstly, the original Schittenhelm algorithm to solve a static assignment problem can be written as follows:

Step 0 (initialisation):

Perform all-or-nothing assignment to the minimum cost routes, p^*_{od} for all origin-destination (*od*) pairs; set $\mathrm{M}^{od} = \{p^*_{od}\}$.

Step 1 (equilibration):

For each origin-destination pair *od* repeat the following steps.

1.1 Find the minimum cost route p^*_{od} based on the current costs.

1.2 If $p^*_{od} \notin M^{od}$, set $\mathrm{M}^{od} = \mathrm{M}^{od} \cup \{p^*_{od}\}$.

1.3 Find the maximum cost route $\overline{p}_{od}$ from M^{od} (i.e. $\overline{p}_{od} = \arg\max_{p \in M^{od}, f_p > 0} C_p$).

1.4 Transfer some flow from $\overline{p}_{od}$ to p^*_{od} so as to equilibrate costs between $\overline{p}_{od}$ and p^*_{od}. This flow transferred from $\overline{p}_{od}$ to p^*_{od} can be decided by the one which minimises Beckmann's (1956) objective function, $\sum_{p \in R_{od}} \int_0^{e_a} c(w)dw$, while keeping other route flows fixed.

1.5 Repeat Step 1 for the next origin-destination pair.

Step 3 (convergence test):

If a convergence criterion ρ_d is less than some test value ε, stop; otherwise, go to Step 2.

Alternatively, here we suggest performing Step 1.4 without optimisation of Beckmann's objective function. The present study has found that we can equilibrate flow pattern between the minimum cost route and maximum cost route by making a single transfer of flows with an amount based on interpolation.

The goal of transferring flows from the maximum cost route to the minimum cost route is to equilibrate costs between the two routes (in Step 1.4). If the cost of p^*_{od} is still less than $\overline{p}_{od}$ after transferring all flows from $\overline{p}_{od}$ to p^*_{od}, all flows should be transferred. Otherwise, we can estimate the optimal transfer using the cost differences between $\overline{p}_{od}$ and p^*_{od}. This can be written mathematically after suppressing the subscript *od* for simplicity:

if $C(p^*)_{\lambda=1} < C(\overline{p})_{\lambda=1}$

$$\lambda^n = 1.0$$

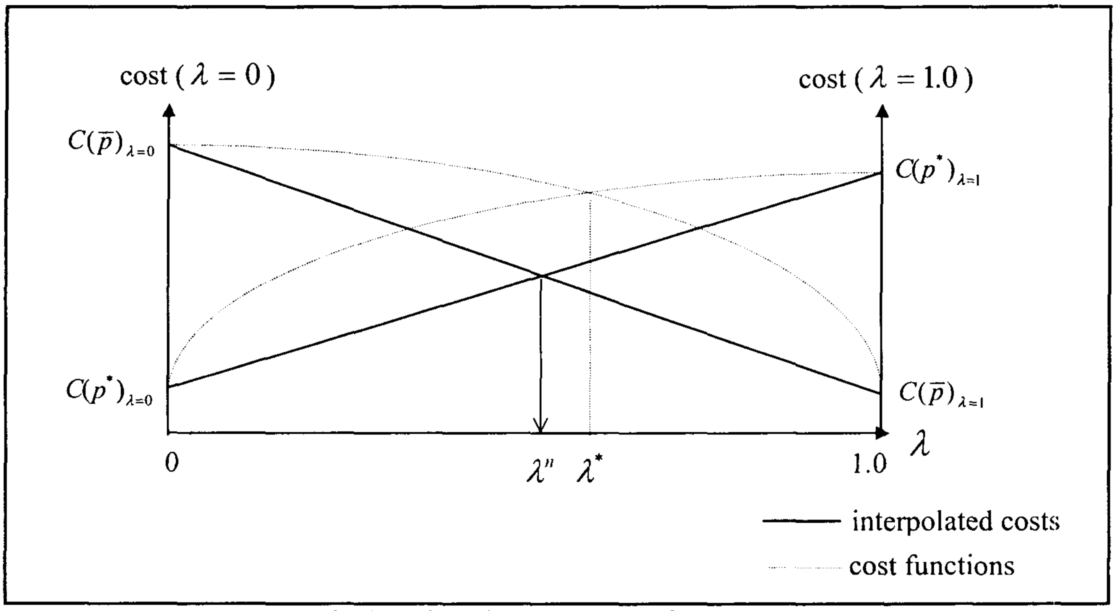
else

$$\lambda'' = \frac{C(\overline{p})_{\lambda=0} - C(p^*)_{\lambda=0}}{(C(\overline{p})_{\lambda=0} - C(p^*)_{\lambda=0}) - (C(\overline{p})_{\lambda=1} - C(p^*)_{\lambda=1})} \qquad (13)$$

where,

λ'' is the proportion of $f_{\overline{p}}$ that is transferred from route $\overline{p}$ to route p^*

$C(\overline{p})_{\lambda=0}$, $C(p^*)_{\lambda=0}$ are the costs of the maximum cost route and the minimum cost route



respectively, when there is no transfer

$C(\overline{p})_{\lambda=1}$, $C(p^*)_{\lambda=1}$ are the costs of the maximum cost route and the minimum cost route respectively, when all flows are transferred from $\overline{p}$ to p^*

Equation (13) can be obtained from the assumption that the cost difference between routes depends on the flow pattern, and hence on the size of the transfer λ. We can represent this relationship graphically as shown in Figure 2:

Figure 2 Optimal flow transfer rate

We can get the following equations from linear interpolation of cost functions:

$$\frac{(C(\overline{p})_{\lambda=1} - C(p^*)_{\lambda=1})}{(C(\overline{p})_{\lambda=0} - C(p^*)_{\lambda=0})} = \frac{1.0 - \lambda^n}{\lambda^n}$$

$$\therefore \lambda^n = \frac{C(\overline{p})_{\lambda=0} - C(p^*)_{\lambda=0}}{(C(\overline{p})_{\lambda=0} - C(p^*)_{\lambda=0}) - (C(\overline{p})_{\lambda=1} - C(p^*)_{\lambda=1})}$$

However, in Figure 2, we can see that the interpolation method will not necessarily identify the optimal value λ^* of λ exactly because route costs do not necessarily change linearly.

This novel algorithm can be applied to solve a dynamic traffic assignment problem (or to solve Step 1 in the triangularisation method) as follows:

Step 1.0 (initialisation of time increment): Set t=0.

Step 1.1 (initialisation of iteration counter): Set n=1.

Step 1.2 (update): Calculate the link travel costs, $\mathbf{c}^n(t+\Delta t)$, using $\mathbf{e}^n(s)$

for $s \le t$, $\mathbf{e}^{n-1}(s)$ for $s > t$.

Step 1.3 (column generation):

Find the minimum cost route $p^*_{od}(t)$ based on the current costs $\mathbf{c}^n(t+\Delta t)$

If $p^*_{od}(t)$ does not belong to the current route set $\mathrm{M}^{od}(t)$, include it with $\mathrm{M}^{od}(t)$ (i.e. $\mathrm{M}^{od}(t) = \mathrm{M}^{od}(t) \cup \{p^*_{od}(t)\}$).

Step 1.4 (equilibration):

Find the maximum cost route $\overline{p}_{od}(t)$ from $\mathrm{M}^{od}(t)$

(i.e. $\overline{p}_{od}(t) = \underset{p \in M^{od}(t),\, f_p(t)>0}{\arg\max}\; C_p(t)$).

Perform network loading after transferring all flows from $\overline{p}_{od}(t)$ to $p^*_{od}(t)$

if $C(p^*_{od}(t))_{\lambda=1} < C(\overline{p}_{od}(t))_{\lambda=1}$

$\overline{\lambda}(t) = 1.0$

else

$$\bar{\lambda}(t) = \frac{C(\bar{p}_{od}(t))_{\lambda=0} - C(p^{*}_{od}(t))_{\lambda=0}}{\{C(\bar{p}_{od}(t))_{\lambda=0} - C(p^{*}_{od}(t))_{\lambda=0}\} - \{C(\bar{p}_{od}(t))_{\lambda=1} - C(p^{*}_{od}(t))_{\lambda=1}\}} \tag{14}$$

Perform network loading after transferring flows based on estimated $\bar{\lambda}(t)$.

Repeat Step 1.3 and 1.4 for all origin-destination pairs.

Step 1.5 (convergence test for inner iteration): if n has reached a pre-specified number or satisfies the convergence criteria, set $t=t+1$ and go to Step 1.1; otherwise, set $n=n+1$ and go to Step 1.2

The advantage of this novel algorithm over the F-W algorithm can be summarised in two points, particularly when they are used to solve dynamic assignment problems. Firstly, the novel solution algorithm can maintain flow propagation automatically when updating the flow pattern since it finds link flows after dynamic network loading with updated route flows. Note that this advantage can be enjoyed in any route-based solution algorithm. Secondly, it does not require any evaluation of the objective function, which would be expensive in the dynamic assignment problem.

5. NUMERICAL EXAMPLE

5.1 Link performance function

In the present dynamic assignment model, the deterministic queuing model is adopted as a link performance function. According to the deterministic queuing model, traffic travels freely along links and then possibly incurs delay at the downstream end, if there is a queue. While inflow is less than capacity of 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 as follows:

$$\frac{dL}{dt} = \begin{cases} 0 & (L_a(t) = 0, \quad e_a(t - \phi) < Q_a) \\ e_a(t - \phi_a) - Q_a & \text{(otherwise)} \end{cases} \tag{15a}$$

$$g_a(t) = \begin{cases} e_a(t - \phi_a) & (L_a(t) = 0, \quad e_a(t - \phi_a) < Q_a) \\ Q_a & \text{(otherwise)} \end{cases} \tag{15b}$$

$$d_a(t) = \frac{L_a(t + \phi_a)}{Q_a} \tag{15c}$$

and travel time (or cost), $c(t)$, can be defined by :

$$c_a(t) = \tau_a(t) - t = \phi_a + d_a(t) \tag{15d}$$

Where,

$L_a(t)$ is the queue length in link a at time t

$d_a(t)$ is the delay in link a incurred by a vehicle that enters the link at time t

Q_a is the capacity for link a

ϕ_a is the free-flow travel time in link a

$\tau_a(t)$ is the time when a vehicle exit the link a after entering at time t

Compared to other link performance functions such as whole link-based outflow model which has been adopted elsewhere in the literature such as Vythoulkas (1990), Wie et al (1990), and Ran and Boyce (1996), this deterministic queuing model is simple and maintains causality, the First-In-First-Out discipline, and correct flow propagation. More detailed discussion on the dynamic traffic modelling can be found in Heydecker and Addison (1998), Astarita (1996), and Han (2002).

5.2 Route travel cost

In static assignment, route travel costs are calculated directly by summing link travel costs along the links that constitute a route, because it is assumed that travel cost is constant regardless of time. However, in the dynamic case, link flows and corresponding link travel costs are not necessarily constant over time. In other words, we take into accounts the fact that the network condition can vary as travellers traverse a route. In this respect, we calculate a route travel cost in mathematical form as:

$$C_p(t) = c_{a_1}(t) + c_{a_2}(t + c_{a_1}(t)) + \cdots\cdots + c_{a_m}(t + c_{a_1}(t) + c_{a_2}(t + c_{a_1}(t)) + \cdots\cdots + c_{a_{m-1}}) \qquad (16)$$

Where,

$C_p(t)$ is the ideal travel time for route p at time t

$c_a(t)$ is the cost of link a at time t and $a_1, a_2, \cdots, a_m \in p$

Note that we sum the cost of each link at the entrance time to that links in order to reflect actual travel costs.

5.3 Two-link network

To examine how the novel solution algorithm works for dynamic assignment within the framework of triangularisation, it has been applied to a simple two-link network, which has a single origin-destination pair. Then, the result is compared to that from the F-W algorithm adopting the objective function as (12). Finally, the flow patterns resulting from both predictive and reactive assignments are compared in the sense of Heydecker and Verlander (1999). Note that an iteration number n in the subproblem of the triangularisation is fixed as 1 according to Sheffi's (1985) streamlined procedure. He showed that it is sufficient to perform iteration once when we solve the subproblem of diagonalisation method.

Table 1 shows the specification of the two-link network and Figure 3 shows a demand profile, which increases at a constant rate $\dot{w}$ until time 10 (min), and maintains peak of $10w$ (veh/min) until time 15 (min), then decreases to zero at a constant rate until time 30 (min).

Table 1 Specification of routes in the two-link network

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

Figure 3 Demand profiles

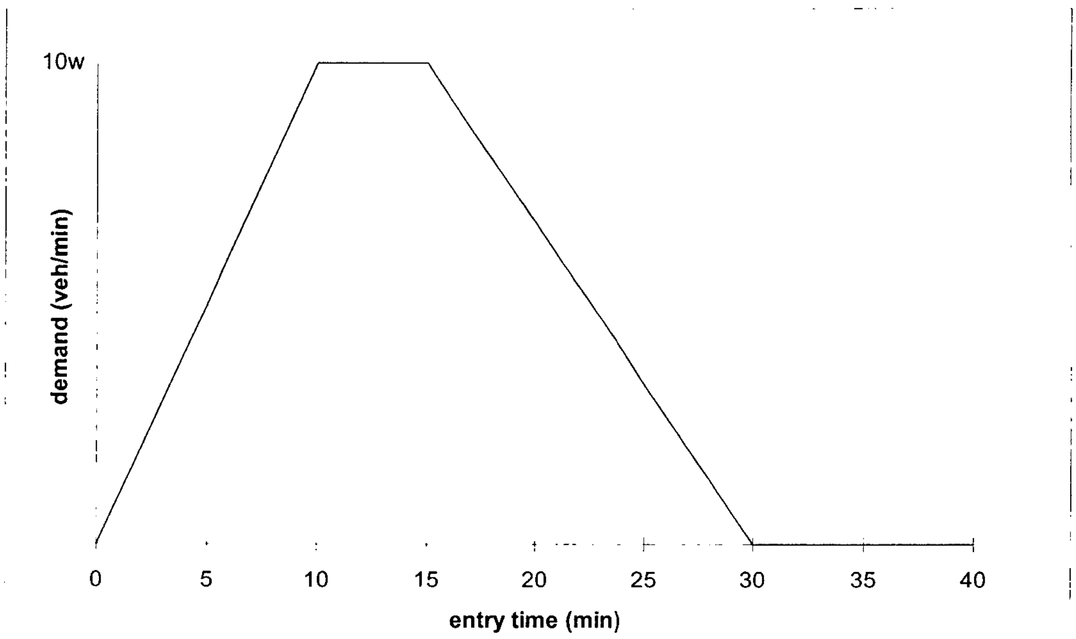


Figure 4 shows the equilibrium travel cost pattern for two routes when we apply the novel solution algorithm in the framework of triangularisation using a time increment of 1 minute and w=5.0 in the demand profile or Figure 3. Note that travel costs are identical for both routes from t=8 to t=28. However, this equilibrium result can be obtained not only from the novel solution algorithm, but also from the F-W algorithm.

Figure 4 Equilibrium travel cost pattern from the novel solution algorithm

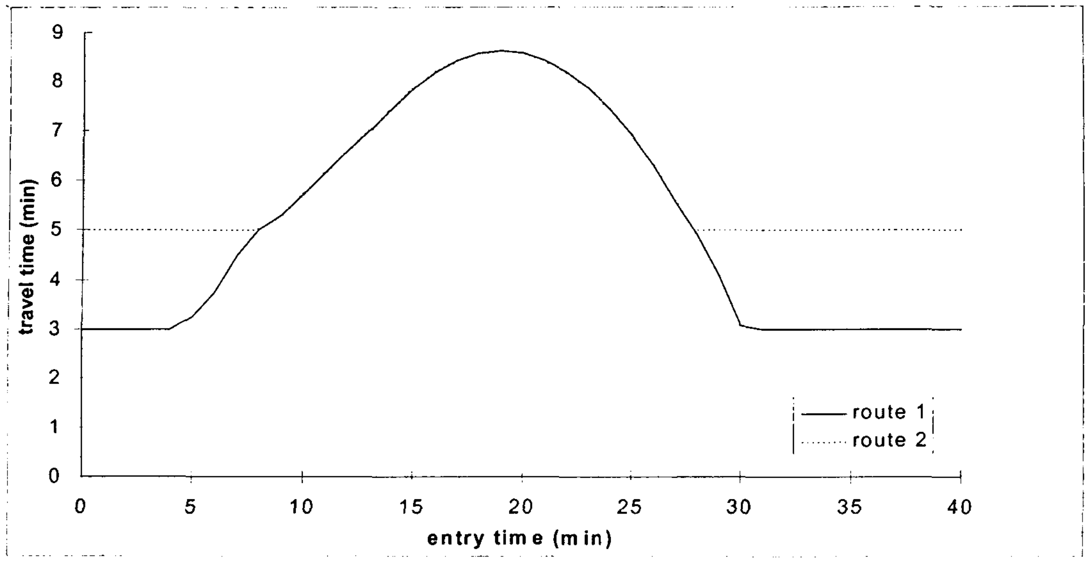


Figure 5 shows that route 2 comes into use from time t=8 when the travel time for route 1 is identical to the free-flow travel costs of route 2, and the route 2 is out of use from time t=28 in the equilibrium assignment. We can see slight difference in assignment proportions between the novel solution algorithm and the F-W algorithm. However, we can note more significant difference between two solution algorithms in terms of the quality of solutions.

Figure 5 Assignment proportion for route 1

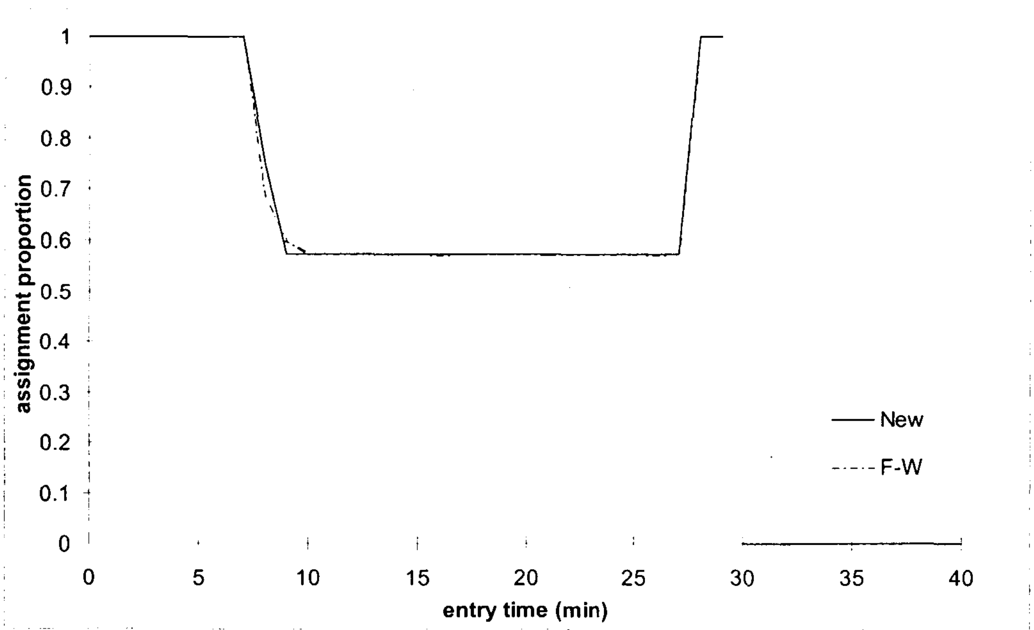


Table 2 shows the measure of disequilibrium over iterations when each of the novel algorithm and the F-W algorithm is applied in the two-link network. The measure of disequilibrium (ξ) is calculated as:

$$\xi = \frac{\sum_t \sum_{od} \sum_p f_p(t) \cdot \left\{ C_p(t) - C^*_{od}(t) \right\}}{\sum_t \sum_{od} \sum_p f_p(t) \cdot C^*_{od}(t)}$$

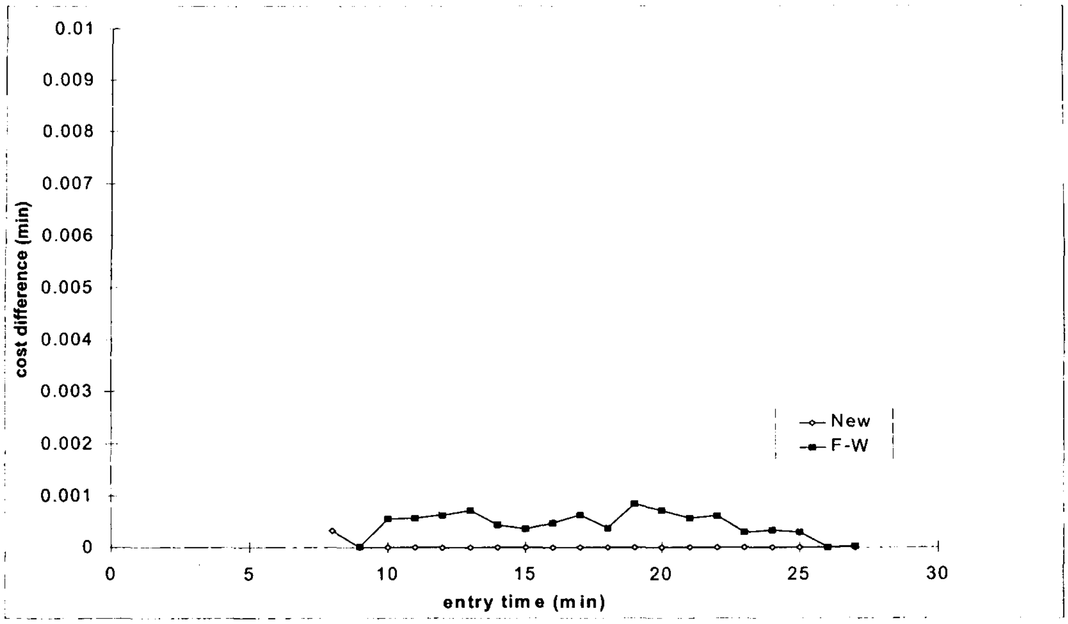
This measure of disequilibrium is derived from Beckmann's complementary inequality (1) and indicates the relative excess cost which is zero in the equilibrium state.

Table 2 Measure of disequilibrium over iterations

Iteration	Novel	F-W
1	1.78127	1.78127
2	0.000002	0.000397
3	$<10^{-6}$	0.000361
4	$<10^{-6}$	0.000349
5	$<10^{-6}$	0.000348
6	$<10^{-6}$	0.000343
7	$<10^{-6}$	0.000337
8	$<10^{-6}$	0.000336
9	$<10^{-6}$	0.000342
10	$<10^{-6}$	0.000348

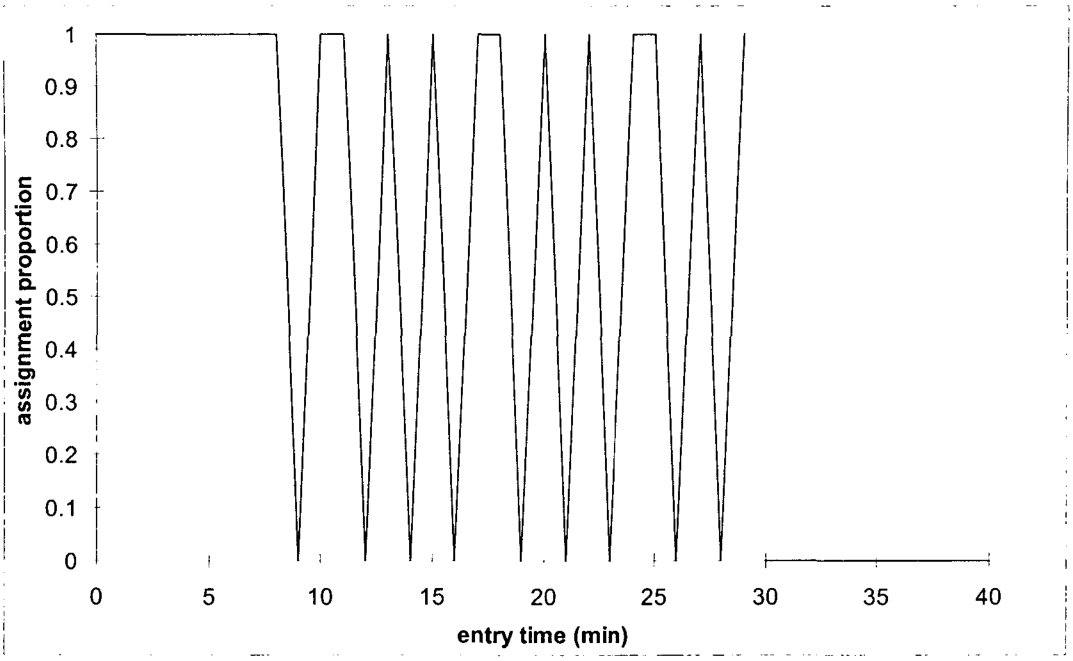
Table 2 shows that the novel solution algorithm gives rise to more accurate equilibrium solution compared to the F-W algorithm, and that the measure of disequilibrium (ξ) will not be improved after iteration 2 in the case of the F-W algorithm. Figure 6 compares the absolute cost difference between route 1 and route 2. As we can see, there is no difference in costs between route 1 and route 2 during [8,27] (when travel times are identical on the two routes) in the case of the novel algorithm, whilst, there is little difference in the case of the F-W algorithm during the same time periods.

Figure 6 Cost difference between two routes



The equilibrium solutions that have been presented so far are based on the predictive cost-flow association. However, when the reactive cost-flow association is made, we find the resulting equilibrium assignment pattern shown in Figure 7. As can be seen, it corresponds to an all-or-nothing assignment to one or other of the routes from t=8 (min). Thus, we cannot find identical travel times between two routes during the time when the two routes are used together.

Figure 7 Assignment proportion for route 1 in equilibrium state (reactive)



Another problem of the reactive cost-flow association can be found when we change the size of time increment (Δt). Figures 8 and 9 compare the equilibrium inflow pattern for route 2 in the two-link network as the size of time increment varies as Δ=0.5, Δ=1.0, and Δ=2.0 in the predictive and the reactive cost-flow association respectively.

Figure 8 Inflows for route 2 for various size of the time increment (predictive)

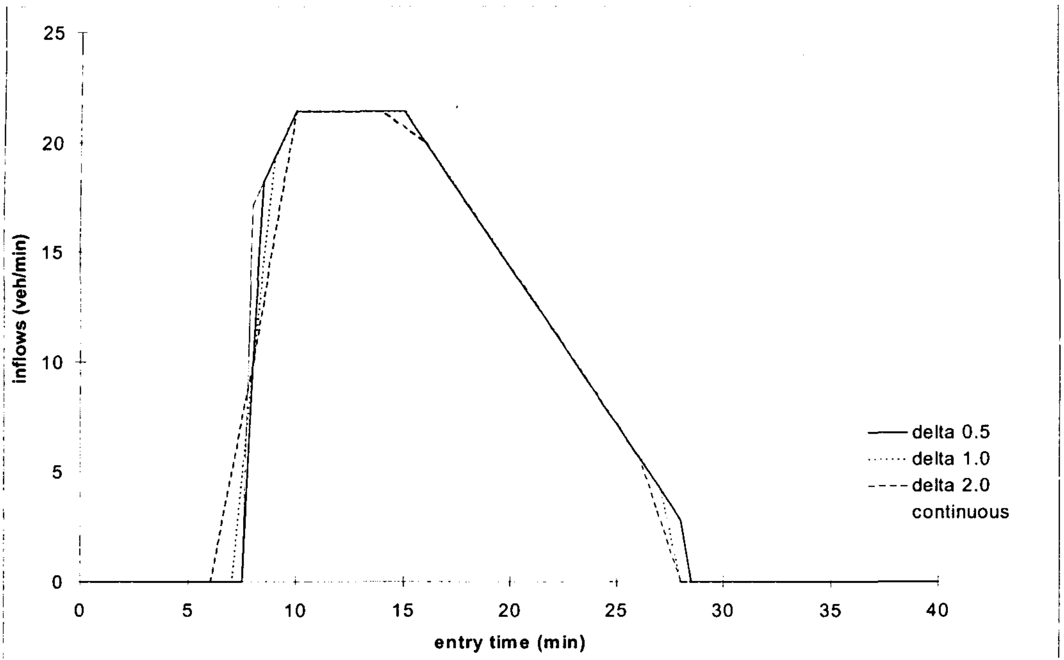
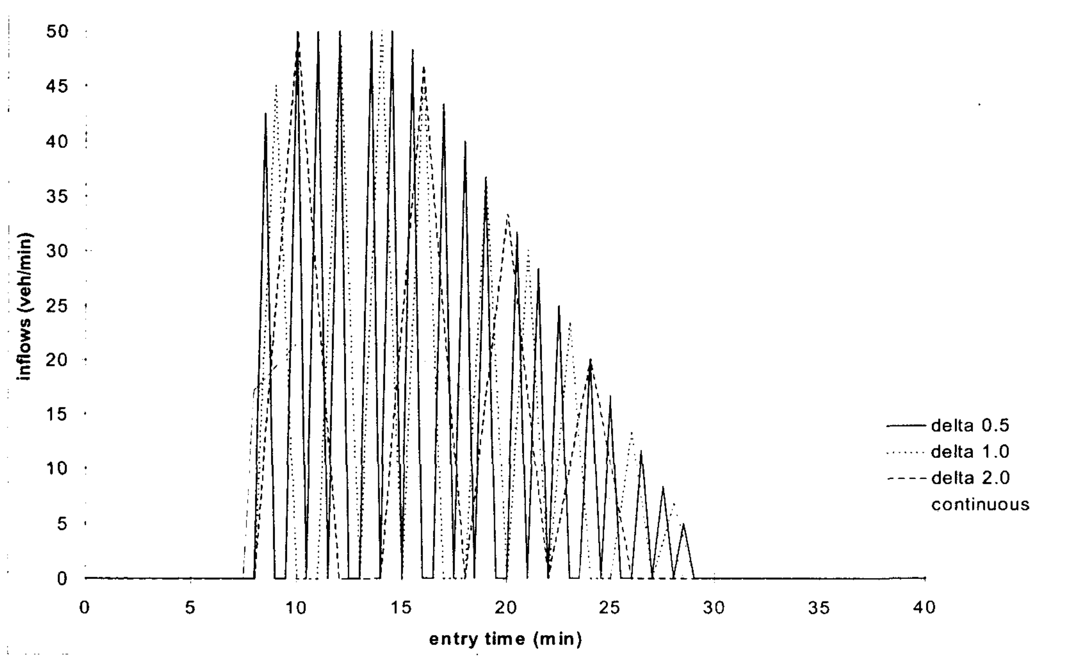


Figure 9 Inflows for route 2 for various size of the time increment (reactive)

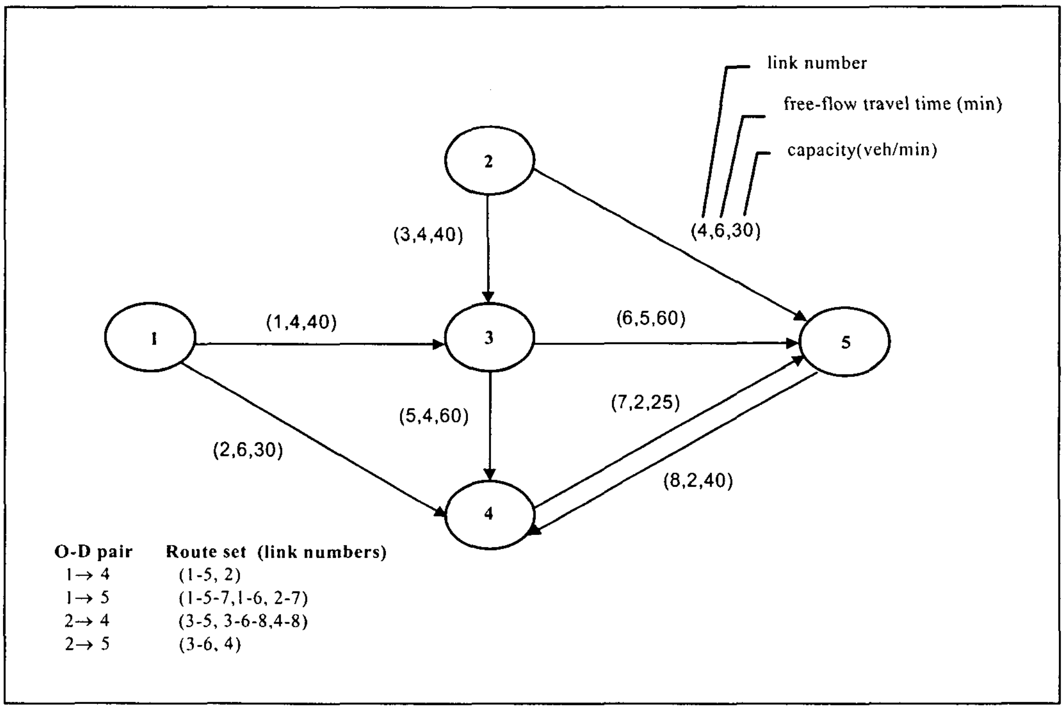


As we can see in Figure 8, there is no substantial difference in the character of the equilibrium inflow pattern for route 2 regardless of the size of time increment in the predictive cost-flow association. Furthermore, the flow pattern becomes similar to that of the analytic solution for continuous time as the size of time increment decreases. This analytic solution is calculated according to the Heydecker and Addison's (1996) novel equilibrium condition. However, we can see from Figure 9 that the inflow pattern changes according to the size of the time increment in the case of the reactive assignment, and this is different in the character from that for continuous time.

5.4 Papageorgiou's network

A more substantial network (Papageorgiou, 1990) is depicted in Figure 10. This network has 5 nodes, 8 links, and 4 origin-destination pairs (1-4, 1-5, 2-4, 2-5). The link number, free-flow travel time, and capacity are shown in the parentheses beside each link. The routes are defined in the lower left-hand side of Figure 10 for each origin-destination pair. These routes are enumerated according to Dial's (1971) reasonable route concept under free-flow travel conditions.

Figure 10 Papageorgiou's network



If we use the same demand profile which was used for the two-link network, we can find the equilibrium solution at iteration 2, and the measure of disequilibrium (ξ) becomes less than 10^{-6} when we apply the new algorithm in the framework of the triangularisation to the test network. Note that we can no longer update link flows by linear combination of the current and the auxiliary flows in Papageorgiou's network, because we should consider flow propagation and because a link is not an automatic route as in the two-link network. Accordingly, the F-W algorithm is not considered in this case, because it needs many network loadings in the line search step and would therefore be computationally demanding.

Figures 11, 12 describe travel cost and assignment patterns at the equilibrium state for origin-destination pairs 1 and 2. Note that only minimum cost routes can carry flows in the user equilibrium state, and this condition has been met for these origin-destination pairs. These good equilibrium solutions were found for the other origin-destination pairs as well.

Figure 11 Equilibrium travel time pattern for OD pair 1

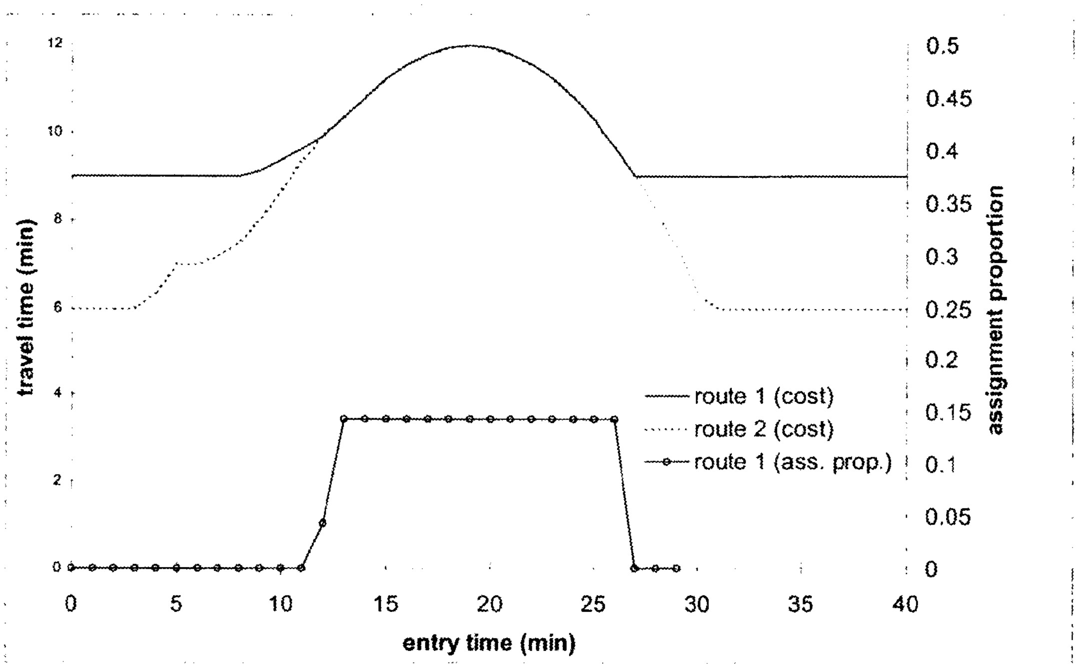
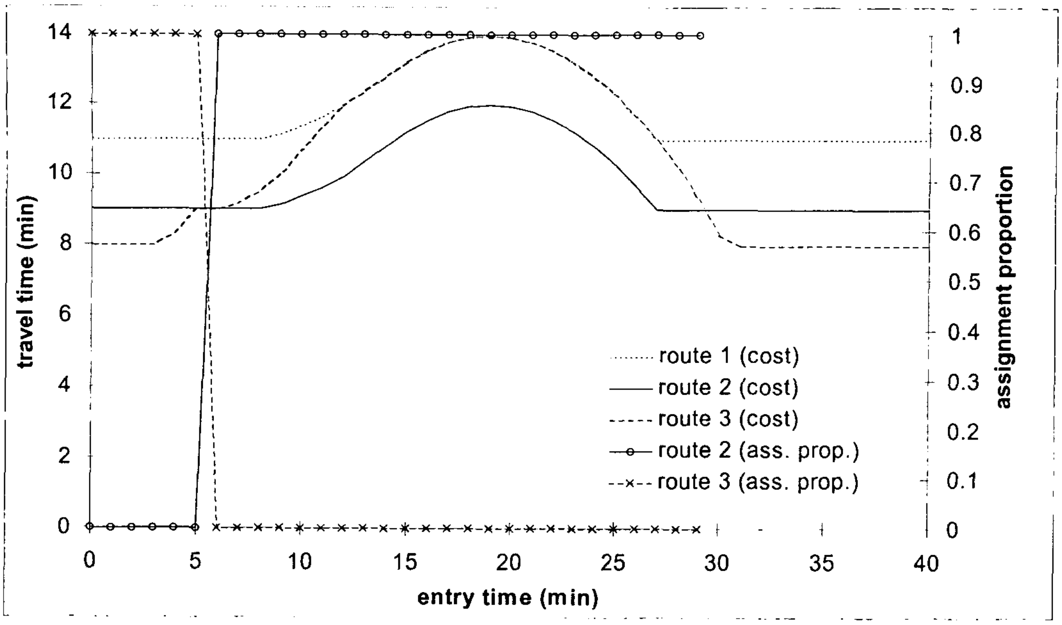


Figure 12 Equilibrium travel time pattern for OD pair 2



As in the case of the two-link network, we can see that only predictive assignment gives rise to plausible assignments in Papageorgiou's network. Figure 13 and 14 describe the inflow pattern for link 3 in the equilibrium state for predictive and reactive cost-flow association respectively as the size of time increment varies as $\Delta=0.5$, $\Delta=1.0$, $\Delta=2.0$.

Figure 13 Inflows for link 3 for various size of the time increment (predictive)

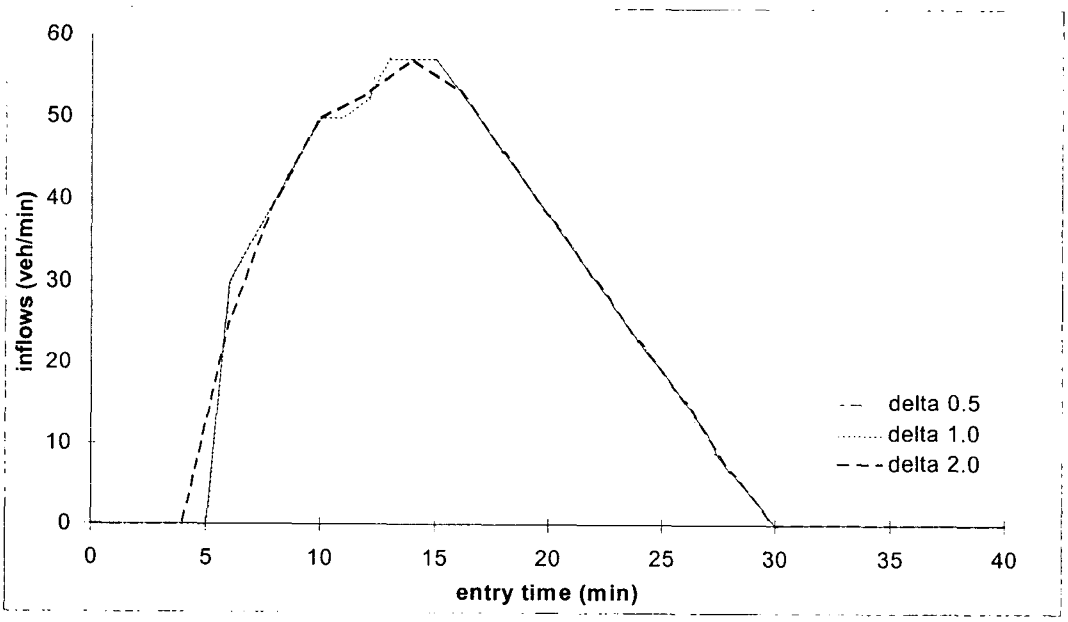
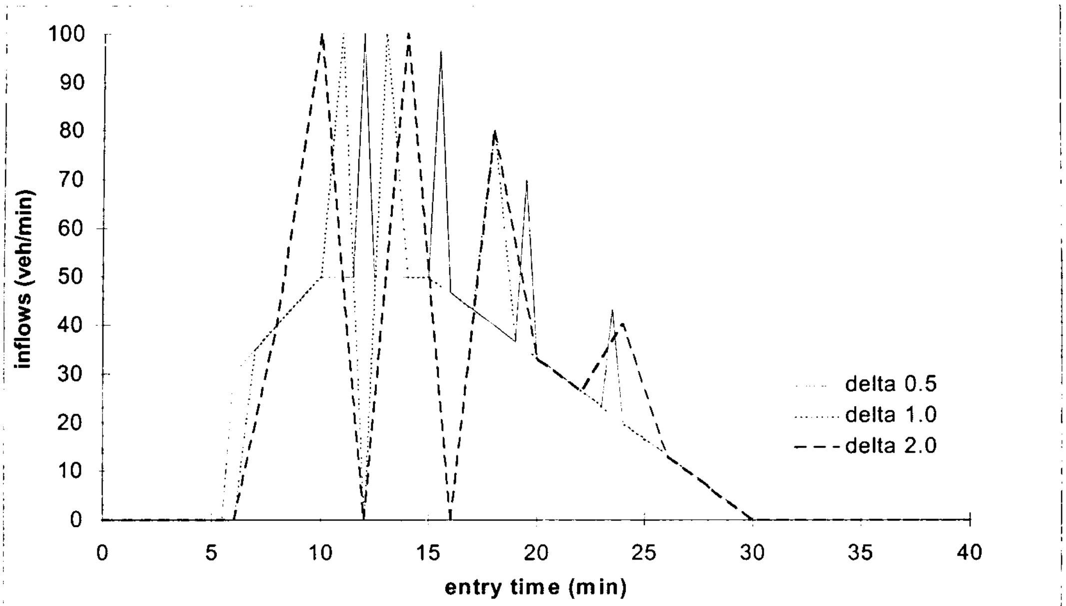


Figure 14 Inflows for link 3 for various size of the time increment (reactive)



We can confirm from Figures 13 and 14 that the flow pattern from the predictive assignment is similar regardless of the size of the time increment, whilst the flow pattern from the reactive assignment changes substantially according to the size of the time increment.

5.5 Sioux Falls network

A more substantial test network, the Sioux Falls network (LeBlanc, 1975) which has 24 nodes and 76 links is shown in Figure 15. In this network we consider 12 origin-destination pairs as Table 3. The specification of the network is summarised in Table 4.

Figure 15 Sioux Falls network

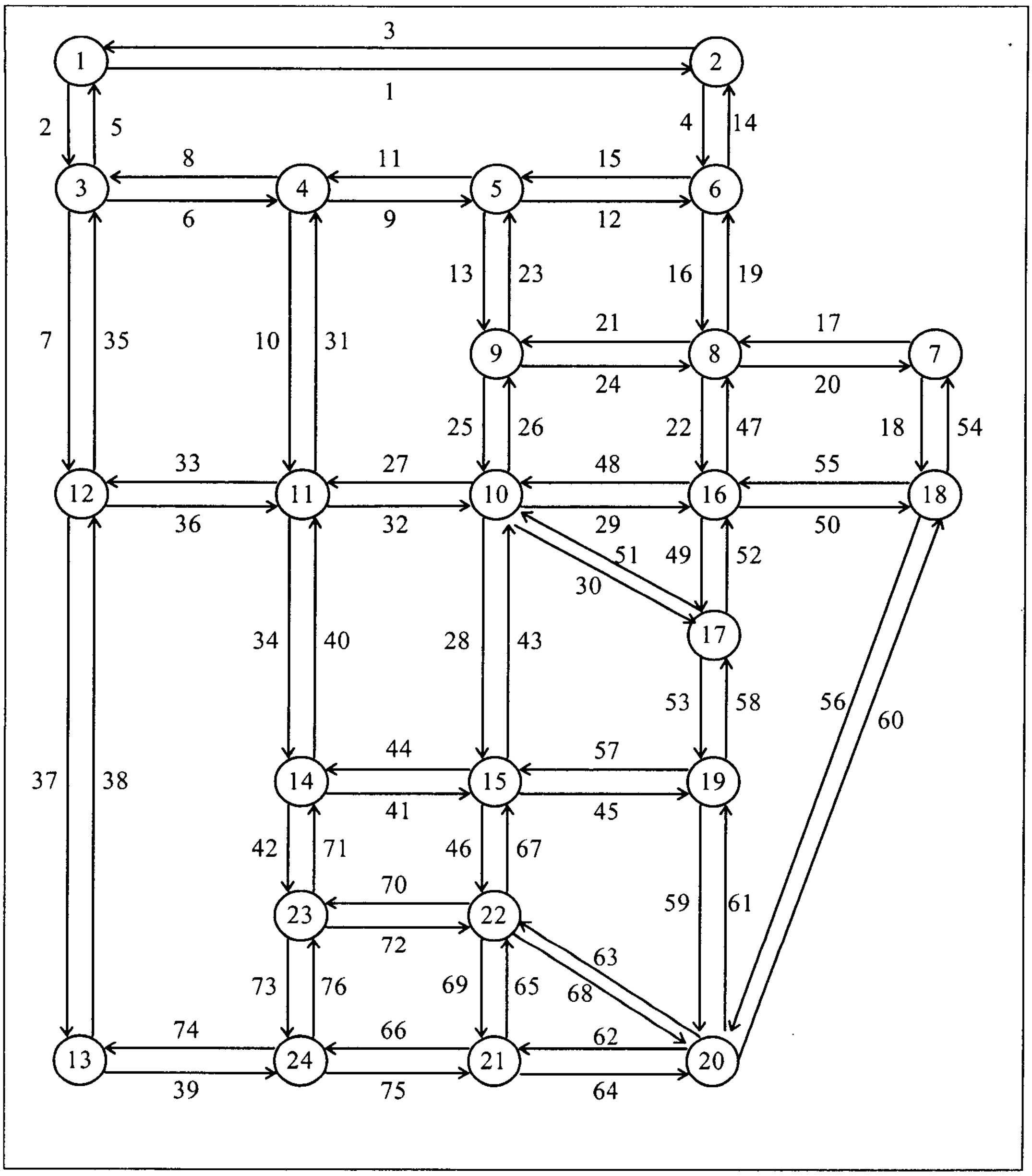


Table 3 New origin-destination pairs for Sioux Falls network

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

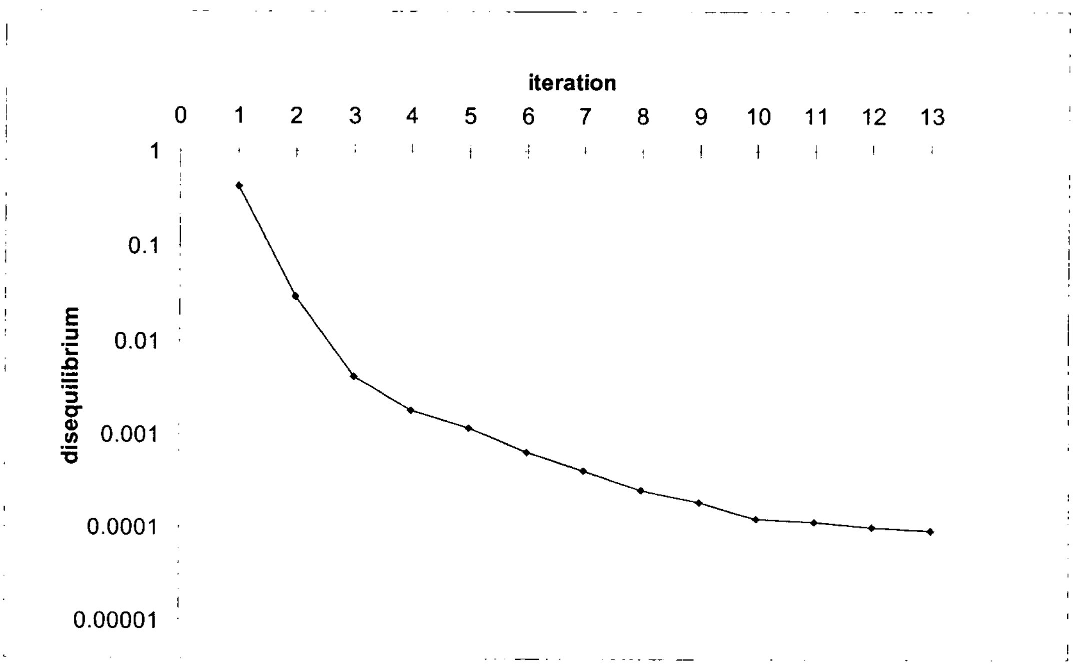
Table 4 Specification of the Sioux Falls network

link	A_node	B_node	free flow travel time (min)	capacity (veh/min)	link	A_node	B_node	free flow travel time (min)	capacity (veh/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	69	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

In the application of dynamic user equilibrium (DUE) assignment to the Sioux Falls network, we just consider the novel solution algorithm because the F-W algorithm cannot maintain correct flow propagation efficiently. Furthermore, we do not consider reactive costs-flows association any more because it does not give rise to plausible results as we have seen in Section 5.3 and 5.4.

The equilibrium solution of the DUE assignment has been found after iteration 13 with the measure of disequilibrium (ξ) as 0.000094 when we take the value of w=3.0 in the demand profile or Figure 3 for all origin-destination pairs, and when we fix the size of time increment (Δt) as 1.0 (min). Figure 16 shows the change of the measure of disequilibrium (ξ) over iteration. We note that ξ decreases monotonically over the iterations.

Figure 16 Change of the measure of disequilibrium (ξ) over iteration



Figures 17, 18, 19, and 20 show the travel cost pattern and assignment proportion for each route for the origin-destination pairs 1-10, 4-19, 6-15, and 7-15 respectively. In those figures, a travel cost pattern and an assignment proportion have been shown without a mark only if the corresponding route carries some flows on it. We can see a good equilibrium travel cost pattern and assignment in all cases. Thus, at each instant only the minimum cost routes carry any flow. This good equilibrium travel cost and flow patterns were found for the other origin-destination pairs as well.

Figure 17 Equilibrium travel cost and assignment for OD pair 1-10

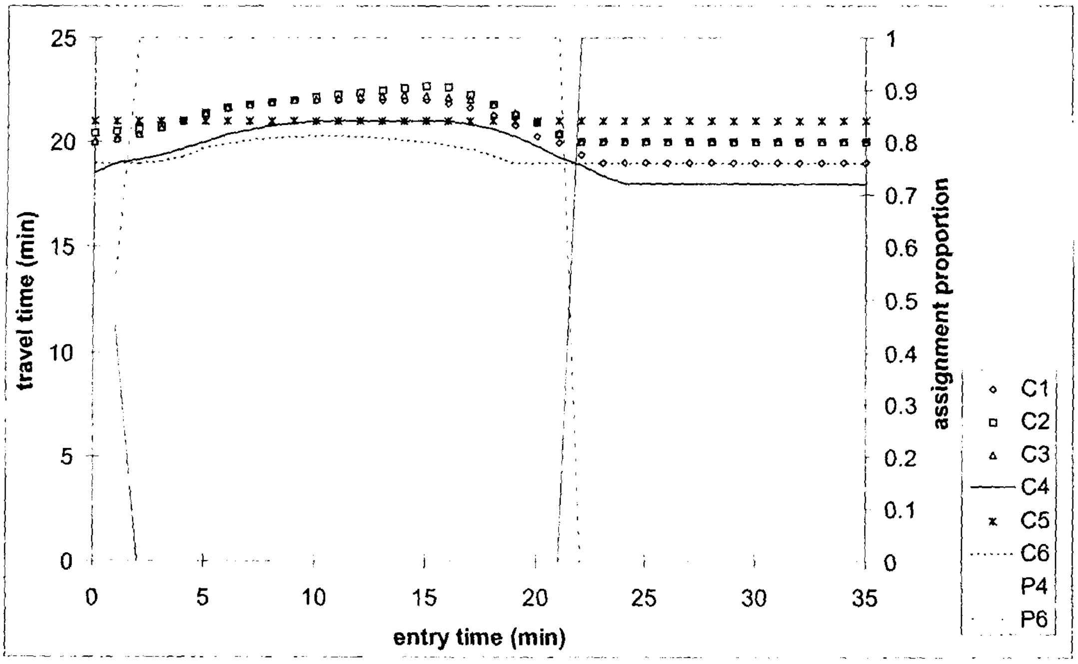


Figure 18 Equilibrium travel cost and assignment for OD pair 4-19

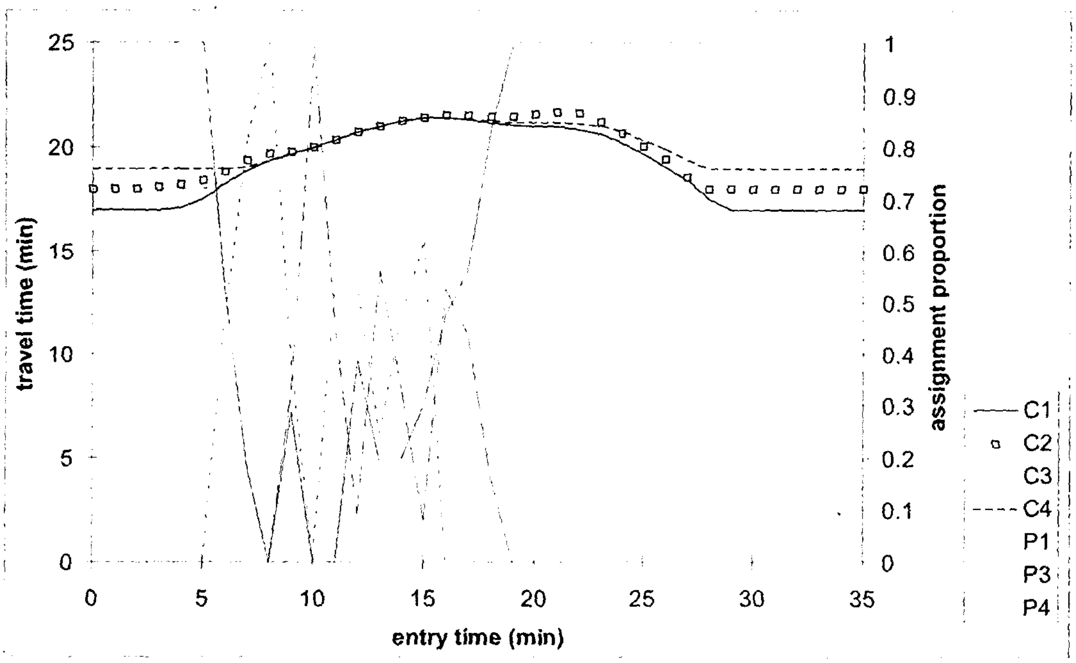


Figure 19 Equilibrium travel cost and assignment for OD pair 6-15

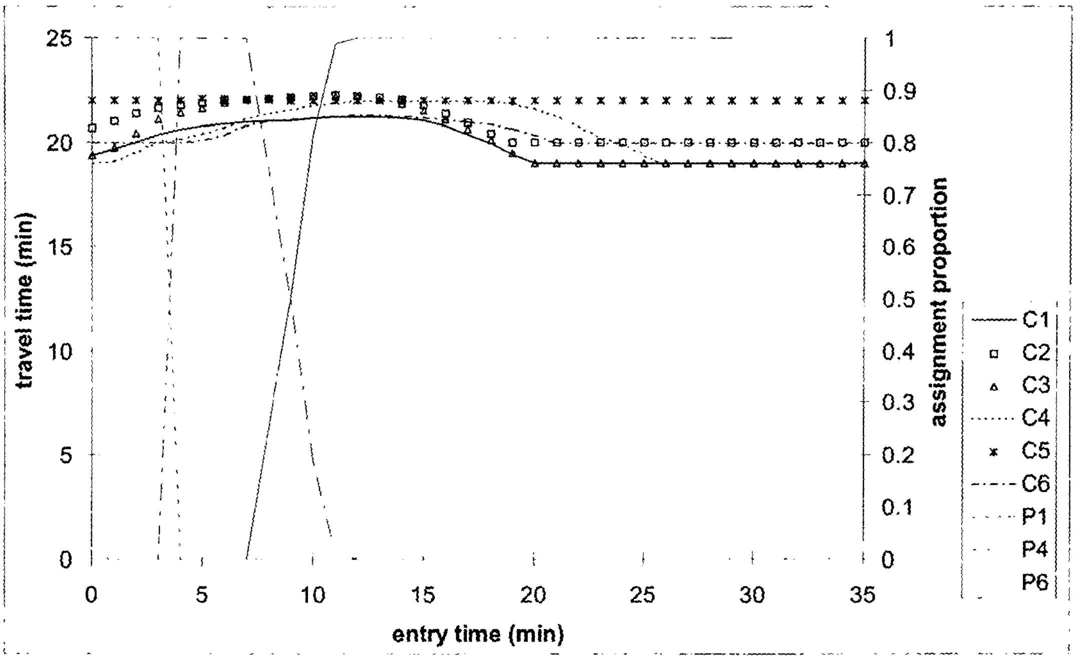
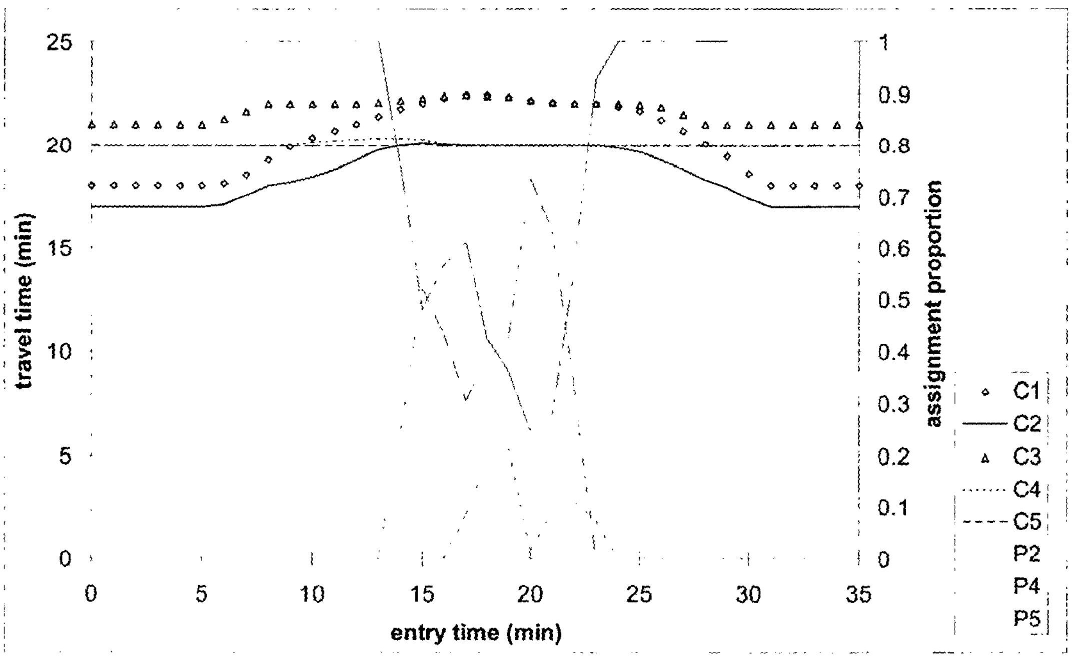


Figure 20 Equilibrium travel cost and assignment for OD pair 7-15



6. DISCUSSION AND CONCLUSION

In the present study, we formulated dynamic user equilibrium (DUE) assignment problem in the form of variational inequality. Then we solved this problem in the framework of the triangularisation method embedding the concept of a forward dynamic programming. To solve the subproblem in the triangularisation method, we developed a novel solution algorithm by modifying Schittenhelm's (1990) algorithm. Furthermore, this study showed a dynamic network loading method that maintains correct flow propagation as well as flow conservation condition.

In particular, this study pointed out that the conventional Frank-Wolfe (F-W) algorithm can be inefficient and inconvenient to solve the dynamic assignment problem because we can no longer update link flows by taking a linear combination of the current and auxiliary flows. This algorithm needs additional dynamic network loading whenever we change the move size in the line search step in order to maintain correct flow propagation. By contrast, the novel solution algorithm, which was developed in this study, can be relatively efficient compared to the F-W algorithm because the novel algorithm can maintain correct flow propagation intrinsically and does not need the evaluation of the objective function.

Results of application of the present DUE assignment model to some test networks using the deterministic queuing model for a link performance function showed that we could find more accurate solution at less computation times with the novel algorithm compared to the conventional the F-W algorithm. Especially, this new solution algorithm gave rise to good quality solutions even in the Sioux Falls network, which is substantially large enough to reflect real network condition. Furthermore, it was found that the predictive cost-flow association but not the reactive one in discrete time could give rise to plausible flow patterns regardless of the size of time interval.

For future studies, it seems to be worthwhile to apply the present DUE assignment model to realistic networks, which have numerous origin-destination pairs, and evaluate various dynamic traffic management measures in such networks. In addition, it is necessary to incorporate travellers' departure time choice models within the present DUE assignment model.

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.

REFERENCE

ADDISON, J.D. and HEYDECKER, B.G., "Traffic models for dynamic assignment", In : Urban Traffic Networks : dynamic flow modelling and control (ed. N.H. Gartner and G. Improta). London : Springer, 213-231 (1995).

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

BECKMANN, M., McGUIRE, C.B., and WINSTEN, C.B., Studies in the economics of transportation. New Haven : Yale University Press, 1956.

BUISSON, C., LEBACQUE, J.P., AND LESORT, J.B., "travel times computation for dynamic assignment modelling", In: Transportation networks : Recent methodological advances (ed. BELL, M.G.H.). Pergamon, 303-17 (1998).

CHEN, S. and HSUEH, C., "A model and an algorithm for the dynamic user-optimal route choice problem," Transportation Research, 32B(3), 219-234 (1998).

DAFERMOS, S.C. "Relaxation algorithms for the general asymmetric traffic equilibrium problem" *Transportation Science*, **16**(2), 231-40, (1982).

DAFERMOS, S.C., and SPARROW F.T., "The traffic assignment problem for a general network", Journal of research of the national *Bureau of Standards* **73(B)**, 91-118 (1969).

DAGANZO, C.F., "Properties of link travel time functions under dynamic loads", *Transportation* Research, 29B(2), 95-98 (1995).

DIAL R.B. "A probabilistic multipath traffic assignment model which obviates path enumeration", *Transportation Research*, **5**(2), 83-111 (1971).

FRANK, M. and WOLFE, P., "An algorithm for quadratic programming", Naval Research Logistics *Quarterly* **3**, 95-110 (1956).

FRIESZ, T.L., BERNSTEIN, D., SMITH, T.E., TOBIN, R.L., and WIE, B., "A variational inequality formulation of the dynamic networks user equilibrium problem", *Operation Research*, **41**(1), 179-91 (1993).

FRIESZ, T.L., LUQUE, J., TOBIN, R.L., and WIE, B. "Dynamic network traffic assignment considered as a continuous time optimal control problem", *Operation Research*, **37**(6), 893-901 (1989).

HAN, S. "Dynamic traffic assignment techniques for general road networks", Ph.D. Thesis, University of London, U.K, August, 2000.

HAN, S. "Dynamic traffic modelling and dynamic stochastic user equilibrium assignment for general road networks", accepted for the publication in Transportation Research Part B (2002).

HEYDECKER, B.G. and ADDISON, J.D., "An exact expression of dynamic traffic equilibrium", In: Transportation and Traffic Theory(ed. J-B Lesort). Oxford: pergamon, 359-83 (1996).

HEYDECKER, B.G. and ADDISON, J.D. "Analysis of traffic models for dynamic equilibrium traffic assignment", In: Transportation Networks: Recent Methodological Advances (ed MGH Bell). Oxford, Pergamon, (1998).

HEYDECKER, B.G. and VERLANDER, N., "Calculation of dynamic traffic equilibrium assignments", Proceedings of the European Transport Conferences, Seminar F, P434, 79-91 (1999).

JANSON, B.N., "Dynamic traffic assignment for urban road networks", *Transportation Research*, **25B**(2/3), 143-61 (1991).

JAYAKRISHNAN, R., TSAI, W.K., and CHEN, A., "A dynamic traffic assignment model with traffic-flow relationships", *Transportation Research*, **3C**(1), 51-72 (1995).

KUWAHARA, M. and AKAMATSU, T., "Decomposition of the reactive dynamic assignment with queues for a many-to-many origin-destination pattern", *Transportation Research*, **31B**(1), 1-10 (1997).

LARSSON, T. and PATRIKSSON, M., "Simplicial decomposition with disaggregated representation for the traffic assignment problem", *Transportation Science* **26**(1), 4-17 (1992).

LAM, W.H. and HUANG, H.J. "Dynamic user optimal traffic assignment model for many to one travel demand", *Transportation Research*, **29B**(4), 243-59, (1995).

LEE, S. Mathematical programming algorithms for equilibrium road traffic assignment. PhD thesis, University of London, London, U.K., 1995.

LEVENTHAL, T., NEMHAUSER G. and TROTTER L., "A column generation algorithm for optimal traffic assignment", *Transportation Science* **7**(2), 168-176 (1973).

LIN, W.H. and LO, H.K. "Are the objective and solutions of dynamic user-equilibrium models always consistent?", *Transportation Research*, **34A**, 137-44, (2000).

MERCHANT, D.K. and NEMHAUSER, G.L., "A model and an algorithm for the dynamic traffic assignment problem", *Transportation Science*, **12**(3), 183-99 (1978).

PAPAGEORGIOU. "Dynamic modelling, assignment, and route guidance in traffic networks", *Transportation Research*, **24B**(6), 471-95 (1990).

PATRIKSSON, M., The Traffic Assignment Problem-Models and Methods, VSP, Utrecht, The Netherlands, 1994.

RAN, B. and BOYCE, D.E., Modelling Dynamic Transportation Networks : An Intelligent Transportation System oriented approach, Springer, 1996.

SCHITTENHELM, H., "On the integration of an effective assignment algorithm with path and path-flow management in a combined trip distribution and traffic assignment algorithm", PTRC European Transport and Planning, 18th Summer Annual Meeting, Transportation Planning Methods, Seminar H, University of Sussex, England, September, 203-214 (1990).

SHEFFI, Y., Urban transportation networks: equilibrium analysis with mathematical programming methods, Prentice-Hall, Englewood Cliffs, New Jersey, 1985.

SMITH, M.J., "The existence, uniqueness and stability of traffic equilibria", *Transportation Research*, **13B**(4), 295-304 (1979).

SMITH, M.J., "A new dynamic traffic model and the existence and calculation of dynamic user equilibria on congested capacity-constrained road networks", *Transportation Research*, **27B**(1), 49-63 (1993).

STEENBRINK, P.A. "Optimisation of transport networks", John Wiley & Sons, London, (1974).

VYTHOULKAS, P., "A dynamic stochastic assignment model for the analysis of general networks", *Transportation Research*, **24B**(6), 453-69 (1990).

WARDROP, J.G., "Some theoretical aspects of road traffic research", Proceedings of the Institution of Civil Engineers, **1**(2), 325-62 (1952).

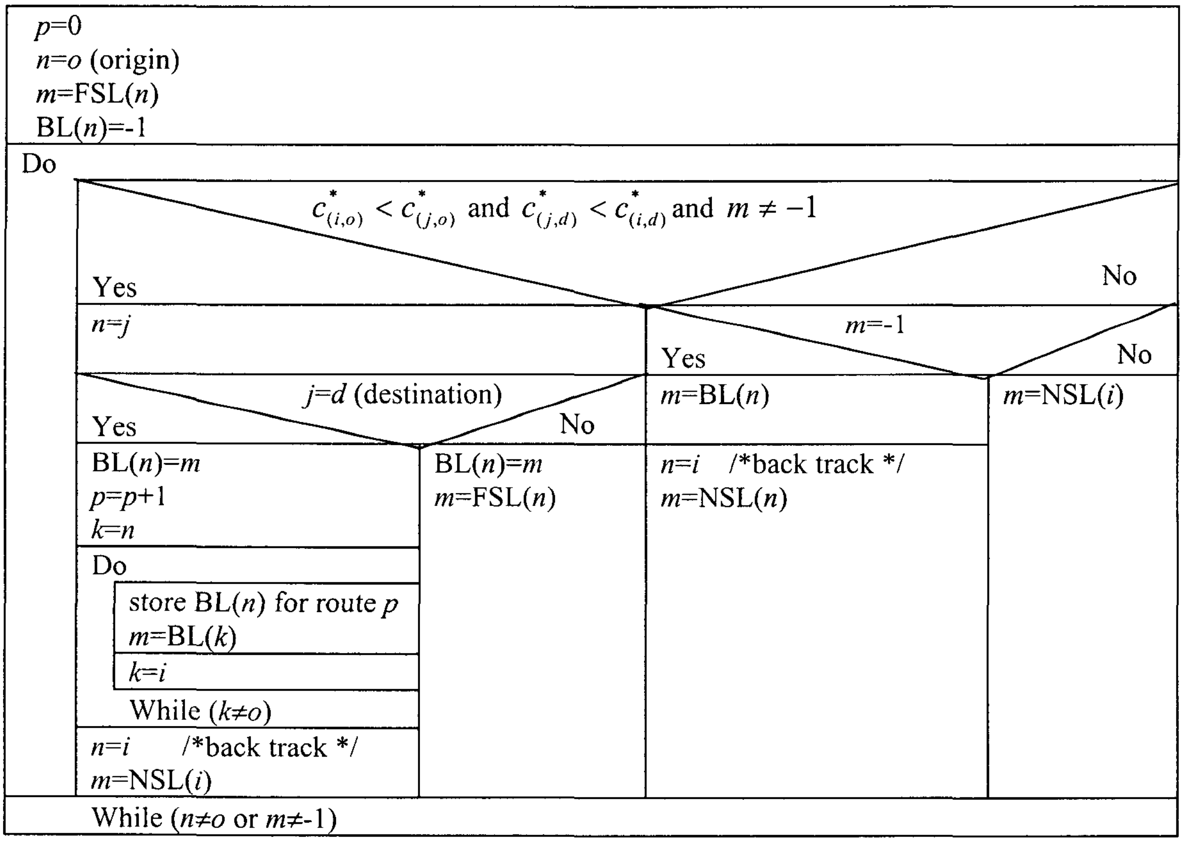
WIE, B., FRIESZ, T.L., and TOBIN, R.L., "Dynamic user optimal traffic assignment on congested multi destination networks", *Transportation Research*, **24B**(6), 431-42 (1990).

WIE, B., TOBIN, R.L.,and FRIESZ, T.L., "The augmented Lagrangian method for solving dynamic network traffic assignment models in discrete time", *Transportation Science*, 28(3), 204-20 (1994).

APPENDIX 1 A BACK-TRACK ALGORITHM

We define the following notations to explain back-track algorithm to enumerate reasonable routes in the sense of Dial (1971):

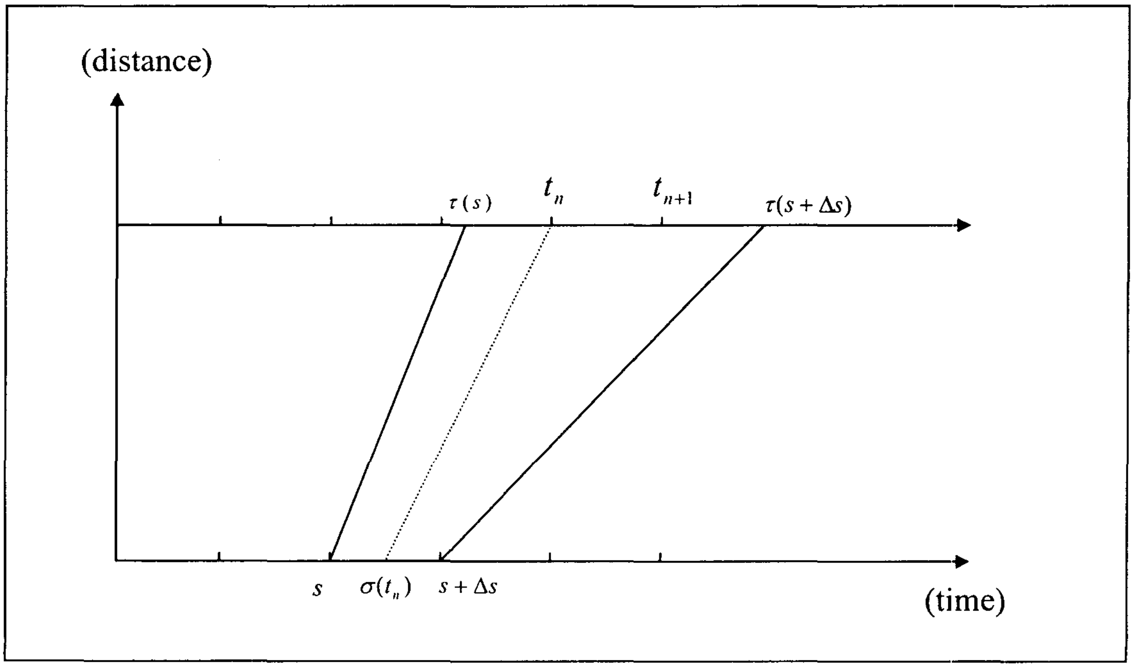
> p: route number
> m: a link which connects node i to j
> FSL (n): the first link starting from node n
> NSL (m): the next link of link m (NSL (m)=-1, if m is the final link starting from node i)
> BL (n): a back link of node n
> $c^*_{(i,j)}$: cost on link (i, j) in the free-flow condition



APPENDIX 2 CALCULATION OF OUTFLOWS IN DISCRETE TIME

Assume that we can draw vehicle trajectories in a link for a certain time increment $[s, s+\Delta s)$ as Figure A-1. Then, we may come across the discrete times t_n, t_{n+1},...,etc., which lies between arrival time $\tau(s)$ and $\tau(s+\Delta s)$. Here we shows how we can calculate outflow rates at these discrete time. Note that for the sake of simplicity we suppress the subscript a, which denotes a link and the subscript od, which denotes an origin-destination pair.

Figure A-1 Vehicle trajectory in a link



Firstly, we can calculate cumulative outflows $G(t_n)$ at time t_n as:

$$G(t_n) = G(\tau(s)) + \bar{g}(\cdot)(t_n - \tau(s)) \tag{A.1}$$

where, $\bar{g}(\cdot)$ denotes outflow rate during $[\tau(s), t_n)$, and is assumed to equal the outflow rate during $[\tau(s), \tau(s+\Delta s))$. Then, we can calculate $\bar{g}(\cdot)$ as:

$$\bar{g}(\cdot) = \frac{G(\tau(s+\Delta s)) - G(\tau(s))}{\tau(s+\Delta s) - \tau(s)} \tag{A.2}$$

Therefore, we can rewrite (A.1) using flow propagation (7) as:

$$G(t_n) = E(s) + \frac{E(s+\Delta s) - E(s)}{\tau(s+\Delta s) - \tau(s)}(t_n - \tau(s)) \tag{A.3}$$

So, we can calculate outflow rate $g(t_n)$ at each discrete time t_n as:

$$g(t_n) = \frac{G(t_{n+1}) - G(t_n)}{\Delta t} \tag{A.4}$$

Although we can calculate outflow rates from a certain link at each discrete time using (A.3) and (A.4), we need to calculate each outflow from various routes which constitute a link from the link outflow in order to perform network loading. For each instant t, this can be calculated using the following equation (A.5) which is proved by Kuwahara and Akamatsu (1997):

$$\frac{h_p(\tau(t))}{g_a(\tau(t))} = \frac{f_p(t)}{e_a(t)} \quad \text{for } \forall p \text{ which uses link } a \tag{A.5}$$

According to the equation (A.5), we can calculate the route outflow which uses link a at discrete time t_n as:

$$\frac{h_p(t_n)}{g_a(t_n)} = \frac{f_p(\sigma(t_n))}{e_a(\sigma(t_n))} \quad \text{for } \forall p \text{ which satisfy } \delta_a^p = 1 \tag{A.6}$$

where $\sigma(t_n)$ is the entering time to a link in order to exit at time t_n and is decided as follows.

Firstly, find k^* which satisfies $\tau(t_n - k\Delta t) \le t_n$ by increasing k from 1.

Let $\bar{s} = t_n - k^*\Delta t$, then we can obtain $\sigma(t_n)$ from interpolation as:

$$\sigma(t_n) = \bar{s} + \frac{t_n - \tau(\bar{s})}{\tau(\bar{s} + \Delta t) - \tau(\bar{s})} \Delta t \tag{A.7}$$

We can also calculate $e_a(\sigma(t_n))$ or $f_p(\sigma(t_n))$ during $[\sigma(t_n), \sigma(t_n) + \Delta t)$ as:

$$e_a(\sigma(t_n)) = e_a[\sigma(t_n)]([\sigma(t_n)] + \Delta s - \sigma(t_n)) + e_a[\sigma(t_n) + \Delta s](\sigma(t_n) - [\sigma(t_n)]) \tag{A.8a}$$

$$f_p(\sigma(t_n)) = f_p[\sigma(t_n)]([\sigma(t_n)] + \Delta s - \sigma(t_n)) + f_p[\sigma(t_n) + \Delta s](\sigma(t_n) - [\sigma(t_n)]) \tag{A.8b}$$

where, $[x]$ denotes the integer part of number x

Note that Δt should be less than travel cost $c(t)$, otherwise we cannot calculate outflow rate after entrance time $s + \Delta s$ properly (see for example, Kuwahara and Akamatsu, 1997). In this respect, they suggested that Δt should be less than the minimum link travel cost as:

$$\Delta t \le \min_a \phi_a, \tag{A.9}$$