An Algorithm for Real-Time User Equilibrium Traffic Assignment

A.E. Haghani[1] and Y. Chuang[2]

Traffic assignment is one of the most important steps of the transportation planning process. Over the years, numerous traffic assignment algorithms have been proposed for predicting the distribution of traffic flow over the network links. The most widely used procedure is perhaps the convex combination method which is used to solve the user equilibrium problem formulated by Beckmann. In the context of IVHS, where we are dealing with real-time information and data processing, this method cannot be implemented due to its lack of consideration of the time factor. This paper proposes a new mathematical formulation of real-time traffic assignment problem which incorporates time into the user equilibrium problem formulation and presents an efficient algorithm for its solution.

In this approach, the overall study period is divided into several variable-length time intervals and a network flow model is formulated which can be solved by successive implementation of the convex combination method. This model is based on a time-space network which is comprised of real and pseudo nodes. The procedure assigns the variable origin-destination demand in each time period over the links of the network according to the existing link flows, network configuration and link performance functions. This algorithm is basically an event-based assignment technique which requires an update of the traffic assignment depending on the conditions of the network and traffic. Two example networks, with seven nodes and ten links, one with single and another with time-varying O/D demands, are presented to illustrate the assignment procedure and to compare the results with the "static" assignment. The results of above example networks demonstrate that the new assignment algorithm is capable of dealing with the real world real-time traffic assignment problems.

INTRODUCTION

Traffic assignment problem is one of the most important steps in transportation planning process. Numerous assignment models and algorithms have been developed over the past several years to distribute the traffic flows over a transportation network. User equilibrium assignment [1] certainly is one of the best models currently used and is based on the assumption that every motorist will try to minimize his or her own travel time when traveling from his/her origin to his/her destination. The Frank-Wolfe method which was originally suggested by Frank and Wolfe as a procedure for solving quadratic programming, was used by LeBlanc et al. [2] to solve the user equilibrium traffic assignment problem.

1. Department of Civil Engineering, University of Maryland, College Park, MD 20742, USA.
2. Viggens Inc., Washington, DC 20024, USA.

Scientia Iranica, Vol. 2, No. 3, © Sharif University of Technology, October 1995.

Most of the existing assignment algorithms are "static" because time variable is excluded in these models. These static assignment algorithms are only able to distribute a single O-D demand matrix at one time. This indicates that the static models are unable to implement the real world assignment task. The development of Intelligent Vehicle-Highway Systems (IVHS) will enable individual drivers to grasp the latest traffic information through different resources and to change their routes with response to the changing traffic conditions. In the next few years, vehicles will be equipped with route guidance equipment which allows drivers to select their routes dynamically. Clearly, the traditional static assignment algorithm will be "crippled" at that time due to its lack of "dynamic" characteristics. A feasible real-time traffic assignment method for dynamic computing and distributing real world traffic flow is, therefore, desired.

The objective of this paper is to develop a real-time user equilibrium traffic assignment model for multiple origins and multiple destinations network, and to present an algorithm for its solution. The new model should enable a user to reasonably distribute traffic flows into the network with response to time-varying traffic conditions. This paper focuses on a mathematical programming approach. The new mathematical formulation, which is based on a dynamic time-space network model, incorporates the time variable into the traditional "static" user equilibrium assignment problem. The validity of the new approach is also examined in the paper. An algorithm for solving the new model, which implements the Frank-Wolfe method successively at every time interval over the entire study period, is also proposed. Two numerical examples are presented to illustrate how the proposed algorithm works and to display how the flows change their routes dynamically reacting to the variations in network traffic conditions. The contributions of this paper are, therefore, introducing an innovative mathematical formulation for a complex problem and presenting a simple and practical algorithm for its solution.

AN OVERVIEW OF PREVIOUS STUDIES

The problem of dynamic traffic assignment has been studied by numerous researchers. Previous studies can be grouped into three major areas:

- Mathematical programming approaches.
- Computer simulation approaches.
- Optimal control theoretic approaches.

According to the assumption of individual routing decisions, there are two classes of problems:

- Dynamic user equilibrium traffic assignment.
- Dynamic system optimization traffic assignment.

Each individual driver tries to minimize his/her own travel time in dynamic user equilibrium assignment problem. On the other hand, in dynamic system optimization assignment problem, every driver is coordinated to minimize the total travel time of the entire transportation network instead of that of the individuals.

Mathematical Programming Approaches

Merchant and Nemhauser [3,4] are the pioneers of the dynamic system optimization traffic assignment problem. They introduced the time variable into the mathematical program prior to other research studies and formulated the problem as a discrete time, nonlinear and non-convex mathematical program corresponding to system optimization in a multiple origins and single destination network. The behavior of their dynamic model was examined under the steady-state assumption and their model was proven to be a proper generalization of a conventional static system optimal traffic assignment model [4].

The algorithmic question of implementing the Merchant and Nemhauser (M-N) model was solved by Ho [5]. He showed that for a piecewise linear version of the M-N model, a global optimum is contained in the set of optimal solutions of a certain linear program. He also presented a sufficient condition for optimality, which implies that a global optimum can be obtained by successively optimizing a sequence of linear programs.

Carey [6] resolved an open question as to whether the M-N model satisfies a constraint qualification, which establishes the validity of the optimality analysis presented in [3,4]. Recently, Carey [7] reformulated the M-N model as a convex non-linear mathematical programming model for least-cost flows on a general congested network in which flows vary over time. The model differs from static network models and from most work on multi-period models because it treats the time taken to traverse each link as varying with the flow rate on the link.

Janson [8] formulated the Dynamic User Equilibrium (DUE) assignment problem as a non-linear mixed-integer problem in terms of path flows in a multiple origins and multiple destinations urban road network. He presented DUE as a temporal generalization of the static user equilibrium problem with additional constraints to insure temporally continuous paths of flow. Meanwhile, he also developed a heuristic procedure that generates approximate solutions to DUE for large networks. Recently, he has presented a link flow formulation and a convergent solution algorithm for the DUE traffic assignment problem [9]. The convergent dynamic algorithm uses methods of linear combinations to find successive solutions of DUE. Janson [10] formulated the DUE problem considering variable departure times and scheduled arrival times. The problem was formulated as a bilevel program in which the solutions must satisfy two simultaneous objectives. These two sub-programs can be solved by an iterative algorithm that consistently converges to solutions which closely satisfy the necessary optimality conditions.

Another mathematical formulation with a totally different concept to the same problem was proposed by Zawack and Thompson [11]. They developed a dynamic time-space network flow model which can be operated by allowing the inputs and outputs of the model to vary over time as well as the network itself.

Computer Simulation Approaches

Yagar [12] presented the first computer simulation model to emulate the traffic assignment based on user equilibrium principle while taking into account both time-varying demand and queue evolution. Yagar also presented a heuristic solution algorithm for dynamic system optimal traffic assignment. His model was recently extended by Van Aerde and Yager [13]. Another computer simulation approach to the problem of dynamic user equilibrium assignment was introduced in Brastow [14]. In his model, exogenous travel demands between origin-destination pairs, which are piecewise-constant functions of time, were transformed into piecewise-constant functions of distance via flow/density relations and the principle of conservation of vehicles.

Optimal Control Approaches

In addition to the aforementioned mathematical and simulation approaches, Luque and Friesz [15] applied optimal control theory to the dynamic traffic assignment problem. They reformulated the Merchant and Nemhauser model as a continuous time optimal control problem. The optimality conditions were derived using the Pontryagin minimum principle [16].

Ran, Boyce and LeBlanc [17] formulated two instantaneous Dynamic User-Optimal (DUO) traffic assignment models for a congested transportation network by using optimal control theory. These models are dynamic generalizations of the static user-optimal model. The equivalence of the solution of the two optimal control programs with DUO traffic flow is demonstrated by proving the equivalence of the first order conditions of the two programs with the instantaneous DUO conditions. This continuous time formulation was solved by Boyce, Ran and LeBlanc [18]. They formulated the program into a discrete time and non-linear programming which can be solved by the penalty method and Frank-Wolfe technique. The same research group [19] presented a bilevel programming formulation of DUO departure time and route choice problem.

The model extended their previous DUO model to the case where both departure time and route choice over a general road network must be chosen. The upper-level program solves the DUO departure time choice problem while the lower-level program solves the DUO route choice program.

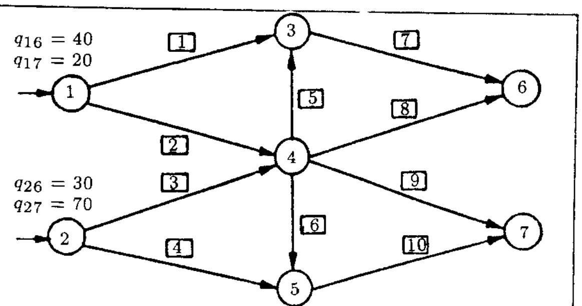
STATEMENT OF THE PROBLEM AND MATHEMATICAL FORMULATION

Real World Traffic Assignment Problem

The traditional UE assignment algorithm only allows users to distribute a single O-D demand trip rate into network links. However, the travel demands from each origin to all other destinations are generated continuously in the real world transportation systems. In other words, the O-D flow rates vary with time. Moreover, the traditional UE assignment algorithm requires that all the links on the network should be empty during its initialization step which may be difficult to achieve when trip rates are variable.

As shown in previous mathematical formulation, Beckmann's UE model implies that flows on the links are spread homogeneously and flow conservation constraints hold only with respect to network configuration (space factor). However, in the real world transportation network, the flow conservation constraints must not only hold in space but also in time. This is the major difference between traditional static UE assignment and "dynamic" traffic assignment.

In order to display how important the time variable is in the assignment problem, the following small network can be examined as an example. Consider a small network with seven nodes and ten links, shown in Figure 1, in which nodes 1 and 2 are origins and nodes 6 and 7 are destinations. The O-D demand from node 1 to node 6, node 1 to node 7, node 2 to node 6 and node 2 to node 7 are 40, 20, 30 and 70 units, respectively. By applying Frank-Wolfe algorithm to solve the static assignment problem, the equilibrium link flows and the associated travel times can be found. The results are also shown in Figure 1. Reviewing the path travel times clearly demonstrates that the flow pattern shown in the figure is the user equilibrium flow due to the equivalency of the path travel times. Examining node 4, the two entering flows 42.721 and 9.761 (link 2 + link 3) should be equal to the four exiting flows 24.231, 15.857, 28.489 and 43.904 (link 5 + link 6 + link 8 + link 9) due to the flow conservation constraint. In the static UE, where the problem is solved for a predetermined period of analysis, this flow conservation may be a good approximation of the real world situation. However, when we are dealing with flow prediction in real-time, we must consider the travel times on various links which enter a node i. Traffic which has entered those links may not arrive at node i at the same time due to variability of travel times over those links. For example, consider node 4 and links 2 and 3. The travel time for these two links are 4.906 and 8.109 time units, respectively. These two



Link number	Link flow	Link travel time
1	17.280	6.165
2	42.721	4.906
3	69.761	8.109
4	30.239	9.653
5	24.231	1.233
6	15.857	1.087
7	41.510	12.698
8	28.489	13.978
9	43.904	23.521
10	46.096	22.197

O-D	Path	UE path travel time
Node 1-6	Link 1+7	18.863
	Link 2+8	18.884
	Link 2+5+7	18.837
Node 1-7	Link 2+9	28.427
	Link 2+6+10	28.190
Node 2-6	Link 3+8	22.087
	Link 3+537	22.040
Node 2-7	Link 3+9	31.630
	Link 4+10	31.850
	Link 3+6+10	31.393

Figure 1. Link flows and travel times of static demand example network.

different travel times clearly indicate that the flow on link 2 arrives at node 4 prior to the flow on link 3 by 3.203 time units. Based on this result, the traffic assignment model without consideration of time is inappropriate and insufficient to be applied for real-time flow prediction. Therefore, there is a need to formulate and solve a real-time flow prediction model which can be implemented in real world transportation system.

Dynamic Time-Space Network Model

The major difference between static and dynamic assignment is whether or not the time factor is included in the model. Intuitively, the dynamic traffic assignment problem can be viewed as incorporation of time variable into the static assignment concept. A very straightforward technique for developing a new assignment concept with dynamic sense is to divide the planning horizon into several time intervals. If static user equilibrium is applicable in each of those time intervals, a new assignment method with dynamic characteristics can be developed by successively applying static user equilibrium algorithm in every time interval. Based on this concept, the constraints which are satisfied in the static assignment formulation, for example nonnegativity, definitional and equality constraints, will still hold in every individual time interval. However, these static UE models must be linked together over time and account for flow entrance and exit over the links of the network.

A dynamic time-space flow network model can be constructed to create a proper environment which allows static user equilibrium algorithm to be successively applied. Conceptually, the status of the time-space network changes with time and new O-D flows that are introduced in the network. The concept of this model is schematically shown in Figure 2. The nodes on the network can be classified into two categories, the real (normal) nodes and pseudo-nodes. The real nodes have the same characteristics as nodes of the original network; on the other hand, the pseudo-nodes and their positions, which are indicated by black color in the figure, represent the position of flows which have entered the link at different points in time and vary over time. The pseudo-nodes are created so that the flows which already exist on the links at the beginning of every time interval could be considered in the static UE model for the next time period. At the beginning of each time period the existing flows over the links can be viewed as flows which are generated at dummy origins (the pseudo-nodes) and are destined to their original destinations. This will enable us to perform the initialization step of the Frank-Wolfe algorithm in the next time interval. Addition of the pseudo-nodes over the links of the original network will create "new" links which connect pseudo-nodes to either real nodes or other pseudo-nodes which are created in previous time periods. Clearly, we have to estimate the volume-delay function of the "new" links, and update those estimates over time.

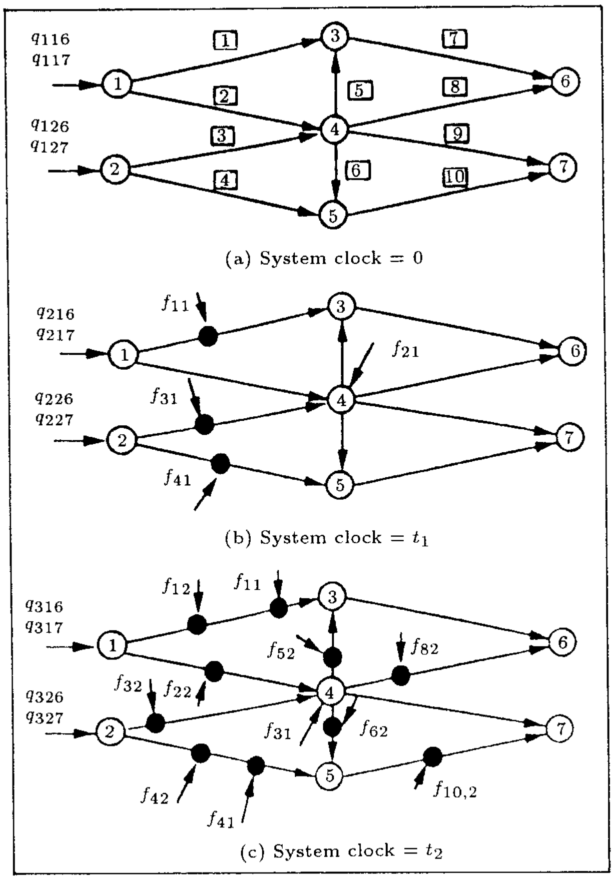


Figure 2. The schematic diagram of dynamic time-space network model.

The schematic diagram of time-space network shown in Figure 2 is used to demonstrate the model more clearly. The network, shown in Figure 2a, with zero system clock is the original network. Nodes 1 and 2 are origins with demands q_{116}, q_{117}, q_{126} and q_{127}, where q_{xyz} denotes flow rates on time interval x from origin y to destination z. Due to the flow conservation constraint, the link flow pattern f_{11}, f_{21}, f_{31} and f_{41} should satisfy $f_{11} + f_{21} = q_{116} + q_{117}$ and $f_{31} + f_{41} = q_{126} + q_{127}$ conditions, where f_{mn} indicates assigned flow on link m at time interval n. As the system clock moves to t_1, the end of first time interval, the network 2 is expanded because of pseudo-nodes, shown in Figure 2b. The "internal" O-D trip rates at those pseudo-nodes for the next time interval (second time interval) are equal to the current flows on each link $(f_{11}, f_{21}, f_{31}, f_{41})$ which were assigned in previous time interval. The new "external" trip rates of the network at the beginning of the second time interval are q_{216}, q_{217}, q_{226} and q_{227}, respectively. Therefore, the new O-D demands matrix for the second time interval can be obtained by integrating the "interval" and the "external" flow rates. The newly expanded network is now ready to apply the static traffic assignment algorithm to the second time interval because the newly expanded O-D demands and an updated network are available. The expanded network and updated O-D demands at the end of the second time interval, when the system clock moves to t_2, are shown schematically in Figure 2c.

This procedure can be iteratively applied through the planning horizon and enables users to predict the link flow pattern in every time interval. Consequently, the time-space model creates a suitable environment for consecutive application of the static user equilibrium algorithm in each time interval and for predicting the UE link flow patterns through the overall planning horizon. It can be shown that the application of this procedure will always result in user equilibrium flows over the network in each time interval [20]. Note that, although it seems that in the implementation of this procedure we need to solve numerous user equilibrium flow problems over successively more complex networks, in reality, for solving each of these problems we can use the final solution of the previous one as an initial feasible solution. Since the status of the network changes only slightly from one iteration to the next, the solution to the successive user equilibrium problems can be obtained extremely quickly and in one or two iterations; therefore, the large number of static user equilibrium assignments presents no significant computational problem. This is further evident from the required solution times for our test networks in which relatively large numbers of user equilibrium assignments were solved. Also note that the procedure internally keeps track of the flows over the links of the network during each time interval and there is no need for keeping track of the location of all flows which are left on the network during these time intervals externally.

Mathematical Formulation for the Real-Time User Equilibrium Traffic Assignment

Basic Assumptions

Since the dynamic time-space network model could create proper environment for consecutive application of the static user equilibrium algorithm in every time interval, the original assumptions of static user equilibrium still hold in this model. For example, all the motorists have full information about network/traffic conditions from various resources (such as in-vehicle driver guidance system or others) have identical behavior and are always able to make correct route choices etc. The following are some more assumptions which are needed to simplify the complicated real-time user equilibrium traffic assignment problem:

- All flow rates should be assigned at desired time.
- External flow rates for every time interval are already known.
- Flows on network links are forced to follow the first-in-first-out sequence.

- The assigned link flow in each time interval is aggregated and viewed as a "cohesive group" of flow.
- Link travel time for a current flow is influenced only by previous flow groups which are still on that link.

Mathematical Formulation

According to the time-space network model and aforementioned assumptions, the real-time traffic assignment problem can be formulated mathematically.

$G(N, A)$ is used to denote a time-space network, where N is the set of nodes and A is the set of links. The planning horizon $[0, T]$ is divided into several time intervals, each of them is indicated by t. For any time interval t we define:

U set of origins.

V set of destinations.

x_a^i aggregated flows entering link a during time interval i.

E_a^{ij} aggregated flows exiting link a during time interval j, which had entered in time interval i.

F_a^{ij} total flows on link a at the end of interval j, which had entered during time interval i and are still on link a.

f_{ki}^{uv} flows on path k between O-D pairs u-v during time interval i.

D_i^{uv} flow rate in the time interval i between an O-D pair u-v.

γ_{aki}^{uv} indicator variable

$= 1$ if link a is on path k between an O-D pair u-v in time interval i.

$= 0$ otherwise.

t_a^{ij} the estimated (remaining) travel time for the flows entered link a in time interval i on/before time interval $j (i \leq j \leq t)$.

$[t_a^{ij}]_{min}$ minimum remaining link travel time (minimum of t_a^{ij}'s).

ϕ_a^{ij} indicator variable

$= 1$ if $i = j$ or flow has entered link a at time interval i and is still on the link at time interval $j (i < j)$

$= 0$ otherwise.

The mathematical formulation for the real-time user equilibrium assignment (RTUE) for the current time interval t, which is in the planning horizon $[0, T]$, is listed below:

$$\min z = \sum_{i=0}^{t} \Big(\sum_a \int_0^{F_a^{it}} t_a^{it}(x) \phi_a^{it} \, dx \Big) , \tag{1}$$

subject to:

$$x_a^i = \sum_u \sum_v \sum_k f_{ki}^{uv} \gamma_{aki}^{uv} \quad \forall a, i , \tag{2}$$

$$E_a^{ij} = x_a^i (1 - \phi_a^{ij}) \quad \forall a, i, j , \tag{3}$$

$$F_a^{ij} = x_a^i - \sum_{m=i}^{j} E_a^{im} \quad \forall a, i, j , \tag{4}$$

$$\sum_k f_{ki}^{uv} = D_i^{uv} \quad \forall u, v, i , \tag{5}$$

$$f_{ki}^{uv} \geq 0 \quad \forall k, u, v, i , \tag{6}$$

$$t_a^{ij}(F_a^{ij})(1 - \phi_a^{ij}) \leq [t_a^{ij}(F_a^{ij})]_{min} \quad \forall a, i, j , \tag{7}$$

$$\phi_a^{(i+1)j} \geq \phi_a^{ij} \quad \forall a, i, j , \tag{8}$$

$$\phi_a^{ij} \geq \phi_a^{i(j+1)} \quad \forall a, i, j , \tag{9}$$

$$\sum_a \sum_i (1 - \phi_a^{ij}) \geq 1 \quad \forall j , \tag{10}$$

$$\forall t \in [0, T], u \in U, v \in V, a \in A .$$

The RTUE program has a non-linear objective function and a convex feasible region. Equation 1 is the objective function of the real-time user equilibrium assignment problem. Similar to the static user equilibrium formulation, the objective function does not have any interpretation and is utilized for mathematical purposes only. The equation is the summation of the objective functions of the static user equilibrium over the planning period considering the flows already on the links and the flows which are being assigned in the current time period. The assigned link flow, f_a, in the static user equilibrium is substituted in this equation by F_a^{it} which is defined above. The program is formulated to minimize the sum of the integral of the travel time functions of the "internal" demands ($F_a^{it} \quad \forall i \in [0,t]$) and the current assigned flow (external) rates ($x_a^t = F_a^{tt}$) under the user equilibrium criteria. Equations 2-10 are the various constraints of the problem. Equation 2 represents the definitional constraints for entering flow on link a at time interval t. These constraints define the relationship between link flows and path flows, and are similar to that of the static user equilibrium formulation. Equation 3 depicts the exit function of the formulation. Based on the assumption made in the previous section, the exiting flow is related to the entering flow and a binary variable which defines whether the entering flow is still on the link. Equation 4 represents the relationship among existing flows, entering flows and exiting flows. Flow conservation constraints in a specific time interval are stated in Equation 5, which guarantees that the flow is conserved in space in any time interval. The nonnegativity constraints, supporting the physical meaning for every flow in every time interval, are presented by Equation 6. Equation 7 defines the relationship between remaining link travel times and their minimum value which is the length of the next time interval and is determined endogenously. For those flows that are still on the links, their remaining travel times are greater than or at least equal to the minimum remaining link travel time of the entire network. If flow F_a^{it} has already left link a in time interval t, then the binary variable ϕ_a^{it} should be equal to zero. The remaining link travel time of this flow is equal to zero due to this constraint. Based on the definition of the binary variable ϕ_a^{it}, the first-in-first-out principle is depicted mathematically by Equations 8 and 9 according to the value of the binary variables ϕ_a^{it}. Equation 10 ensures that at least one flow is going to exit one of the links in the entire network in each time interval.

Discussion

The differences between above formulation and dynamic user equilibrium models as mentioned before are in two aspects. First, the proposed formulation minimizes the objective function of the problem based on every individual flow (including existing link flow groups and flows that are being assigned) instead of considering all of the link flows together. This model takes into account the different entrance and anticipated exit times of the existing flows while estimating the current assigned flows by user equilibrium criteria. Hence, the model avoids overestimation of the travel times of the flows which are being assigned in the most current time period. Secondly, the formulation provides variable time interval concept which ensures that no flow can enter or exit any links of the network in the middle of a time interval. This concept simplifies the problem considerably by ensuring the flow continuity over time.

The mathematical formulation of RTUE problem was shown by Equations 1-10. To show that this formulation satisfies user equilibrium criteria at each time interval, the relational constraints (Equation 4), can be rewritten by being plugged into Equations 2 and 3 as follows:

$$F_a^{ij} = x_a^i - \sum_{m=i}^{j} E_a^{im} = x_a^i - \sum_{m=i}^{j} x_a^i(1-\phi_a^{im})$$
$$= \sum_{uv}\sum_{k} f_{ki}^{uv}\gamma_{aki}^{uv}$$
$$- \sum_{m=i}^{j}(\sum_{uv}\sum_{k} f_{ki}^{uv}\gamma_{aki}^{uv}(1-\phi_a^{im})) \ . \tag{11}$$

The above equation indicates that the existing link flows can be expressed as a function of path flows and binary variables γ_{aki}^{uv} and ϕ_a^{ij}. The mathematical formulation of RTUE program, Equations 1-10, then can be rewritten as follows:

$$\min\ z = \sum_{i=0}^{t}(\sum_{a}\int_0^{F_a^{it}} \phi_a^{it}\ t_a^{it}(x)dx)\ , \tag{12}$$

subject to:

$$F_a^{ij} = \sum_{uv}\sum_k f_{ki}^{uv}\gamma_{aki}^{uv} - \sum_{m=i}^{j}(\sum_{uv}\sum_k f_{ki}^{uv}\gamma_{aki}^{uv}(1-\phi_a^{im})) . \quad (13)$$

and Equations 5-10 as before.

To prove that the above mathematical formulation and the real-time user equilibrium assignment problem are identical, it is necessary to prove that the particular flow pattern which solves the mathematical program also satisfies the equilibrium criteria. Consider the first time interval as the current time interval in which $t = 1$, the objective of this case can be expressed as follows:

$$\min\ z = \sum_a \Big(\int_0^{F_a^{01}} t_a^{01}(x)\phi_a^{01}\, dx + \int_0^{F_a^{11}} t_a^{11}(x)\phi_a^{11}\, dx\Big) , \quad (14)$$

where the equation can be decomposed in terms of F_a^{01} and F_a^{11}. The various constraints can also be broken down in a similar way. Equation 11 is shown as follows:

$$F_a^{01} = \sum_{uv}\sum_k f_{k0}^{uv}\gamma_{ak0}^{uv} - \sum_{uv}\sum_k f_{k0}^{uv}\gamma_{ak0}^{uv}[(1-\phi_a^{00}) + (1-\phi_a^{01})] = \sum_{uv}\sum_k f_{k0}^{uv}\gamma_{ak0}^{uv} - \sum_{uv}\sum_k f_{k0}^{uv}\gamma_{ak0}^{uv}[1-\phi_a^{01}] \quad \forall a , \quad (15)$$

$$F_a^{11} = \sum_{uv}\sum_k f_{k1}^{uv}\gamma_{ak1}^{uv} - \sum_{uv}\sum_k f_{k1}^{uv}\gamma_{ak1}^{uv}(1-\phi_a^{11}) = \sum_{uv}\sum_k f_{k1}^{uv}\gamma_{ak1}^{uv} \quad \forall a . \quad (16)$$

Equation 5 can be decomposed as follows:

$$\sum_k f_{k0}^{uv} = D_0^{uv} \quad \forall u,v , \quad (17)$$

$$\sum_k f_{k1}^{uv} = D_1^{uv} \quad \forall u,v . \quad (18)$$

The non-negativity constraints also can be decomposed as follows:

$$f_{k0}^{uv} \geq 0 \quad \forall u,v,k , \quad (19)$$

$$f_{k1}^{uv} \geq 0 \quad \forall u,v,k . \quad (20)$$

The rest of the constraints are the limitations of the binary variables, ϕ_a^{ij}, which control only the remaining travel times. The overall formulation, therefore, can be divided into two parts. The first portion is the subprogram of the beginning time interval ($i = 0$) and is shown as follows:

$$\min\ z_0 = \sum_a \int_0^{F_a^{01}} t_a^{01}(x)\phi_a^{01}\, dx , \quad (21)$$

subject to:

$$F_a^{01} = \sum_{uv}\sum_k f_{k0}^{uv}\gamma_{ak0}^{uv} - \sum_{uv}\sum_k f_{k0}^{uv}\gamma_{ak0}^{uv}[1-\phi_a^{01}] \quad \forall a , \quad (22)$$

$$\sum_k f_{k0}^{uv} = D_0^{uv} \quad \forall u,v , \quad (23)$$

$$f_{k0}^{uv} \geq 0 \quad \forall u,v,k . \quad (24)$$

The similar subprogram of the second part, which is for the current time interval ($i = 1$), is shown as follows:

$$\min\ z_1 = \sum_a \int_0^{F_a^{11}} t_a^{11}(x)\phi_a^{11}\, dx , \quad (25)$$

subject to:

$$F_a^{11} = \sum_{uv}\sum_k f_{k1}^{uv}\gamma_{ak1}^{uv} \quad \forall a , \quad (26)$$

$$\sum_k f_{k1}^{uv} = D_1^{uv} \quad \forall u,v , \quad (27)$$

$$f_{k1}^{uv} \geq 0 \quad \forall u,v,k . \quad (28)$$

It is clear that both subprograms are identical with the static user equilibrium. The similarity of the decomposition characteristics is maintained when the current time interval moves to $t = 2$. The objective function is shown below when $t = 2$:

$$\begin{aligned}\min z &= z_0 + z_1 + z_2\\ &= \sum_a \int_0^{F_a^{02}} t_a^{02}(x)\phi_a^{02}\ dx\\ &+ \sum_a \int_0^{F_a^{12}} t_a^{12}(x)\phi_a^{12}\ dx\\ &+ \sum_a \int_0^{F_a^{22}} t_a^{22}(x)\phi_a^{22}\ dx\ .\end{aligned}$$

and the definitional constraints are shown as follows:

$$\begin{aligned}F_a^{02} &= \sum_{uv}\sum_k f_{k0}^{uv}\gamma_{ak0}^{uv}\\ &- \sum_{uv}\sum_k f_{k0}^{uv}\gamma_{ak0}^{uv}[(1-\phi_a^{00})\\ &+ (1-\phi_a^{01}) + (1-\phi_a^{02})]\ ,\\ F_a^{12} &= \sum_{uv}\sum_k f_{k1}^{uv}\gamma_{ak1}^{uv}\\ &- \sum_{uv}\sum_k f_{k1}^{uv}\gamma_{ak1}^{uv}[(1-\phi_a^{11}) + (1-\phi_a^{12})]\ ,\\ F_a^{22} &= \sum_{uv}\sum_k f_{k2}^{uv}\gamma_{ak2}^{uv}\\ &- \sum_{uv}\sum_k f_{k2}^{uv}\gamma_{ak2}^{uv}[1-\phi_a^{22}]\ .\end{aligned}$$

The flow conservation and nonnegativity constraints can also be decomposed as discussed previously. Therefore, the static user equilibrium criteria can be matched in every time interval by examining each decomposed subprogram. This indicates that the real-time user equilibrium formulation is able to minimize the objective function of existing link flows, and the O-D demands are being assigned into the network by user equilibrium criteria in each individual time interval.

A REAL-TIME USER EQUILIBRIUM TRAFFIC ASSIGNMENT ALGORITHM

A Real-Time User Equilibrium Assignment Scheme

Under the concept of dynamic time-space network model, the static user equilibrium (Frank-Wolfe algorithm) can be applied at each time interval. However, the algorithm has to be modified to meet the special need of time-space network model. Because the O-D flows vary with time, the link flows must be distinguished and separated according to their initial origins and destinations. In other words, a link flow is the sum of all disaggregate path flows between each O-D which uses that link. The updated algorithm uses these disaggregate path flows to capture the variability of the network with time. In addition to above modification, the following problems should be addressed to develop a new real-time user equilibrium assignment algorithm.

Existing Flow Problem

The static UE algorithm assumes free flow situation for performing all-or-nothing loading procedure during its initialization step. However, flows enter links depending upon the O-D demand rates and link performance functions. It is relatively difficult to ensure that free flow conditions occur at the beginning of each assignment procedure while the O-D demand rate varies over time. The previously mentioned dynamic time-space network model overcomes such difficulties and provides the solution for this problem. The concept of dynamic time-space network is described again hereafter. Consider Figure 3, the demand flow f_{a3} from outside the network is going to be assigned into link a while flows f_{a1} and f_{a2} already exist on the same link. These flows entered the link in previous time periods and, as per our assumptions in the section before last, both of these flows can be viewed as separate "cohesive groups". In order to "clean out" the existing flows and achieve the free flow condition, the positions of these flows can

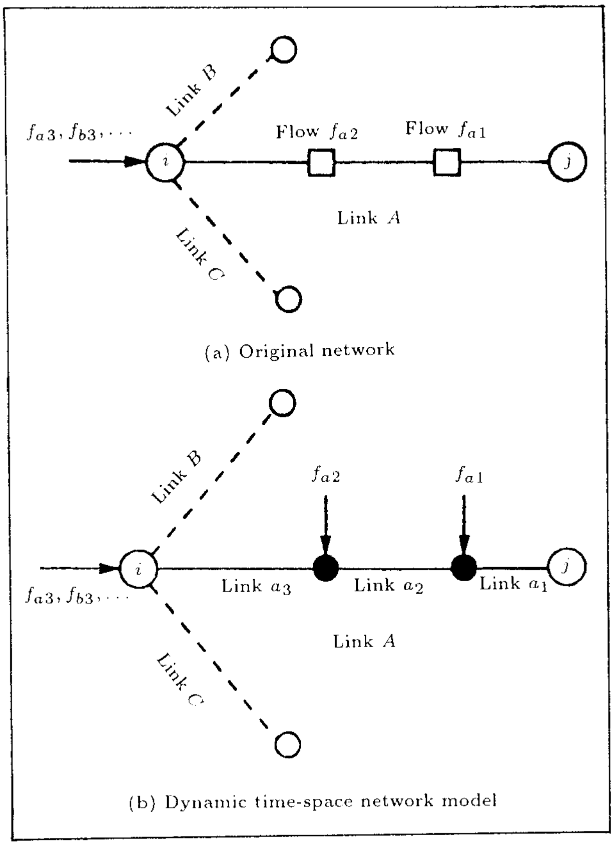


Figure 3. Illustration of dynamic time-space network model.

be treated as two pseudo-nodes and the flows can be seen as "internal" demands from these nodes. The concept is schematically displayed in Figure 3b. The "internal" demands f_{a2} and f_{a1} still keep their original destinations but switch their origin nodes to the current pseudo-nodes A and B, respectively. Following this philosophy, the network itself and the O-D demands, which included "external" demands such as f_{a3} and "internal" demands f_{a2} and f_{a1}, vary with time and enable users to apply static user equilibrium algorithm during each time interval.

Remaining Travel Time Problem

In the dynamic time-space network model, links will be theoretically divided by pseudo or real nodes. Because the pseudo-nodes never become destinations for all the O-D pairs, only the travel times from real or pseudo-nodes to real nodes are needed to be considered in the algorithm. Considering the "new" links a_1, a_2, a_3 and the original link A shown in Figure 3, the "new" link performance functions should be somehow related to that of the original link. As mentioned before, a pseudo-node cannot be a destination. The travel times of flows f_{a1}, f_{a2} and f_{a3} from current positions to next real node j should be estimated in order to determine the flow that is to be assigned next into the link A. The approach for estimating the link travel time for the flow that is going to be assigned is illustrated in Figure 4. The link performance function for link A is $T_a(f_a)$, where f_a denotes the link flow, therefore, the link travel time for the flow f_{a1}, which entered the link at system clock ST_{1_i}, is $T_a(f_{a1})$. As the system clock moves to ST_{2_i}, an undetermined amount of flow, f_{a2}, is going to enter the link. We need to estimate the travel time for flow f_{a2} by taking into account the influence of existing flow f_{a1}. The system clock ST_{1_o} will be equal to $ST_{1_i} + T_a(f_{a1})$ when f_{a1} exits the link. Thus, no other flows are in front of f_{a2} when system clock is ST_{1_o}. From this, the

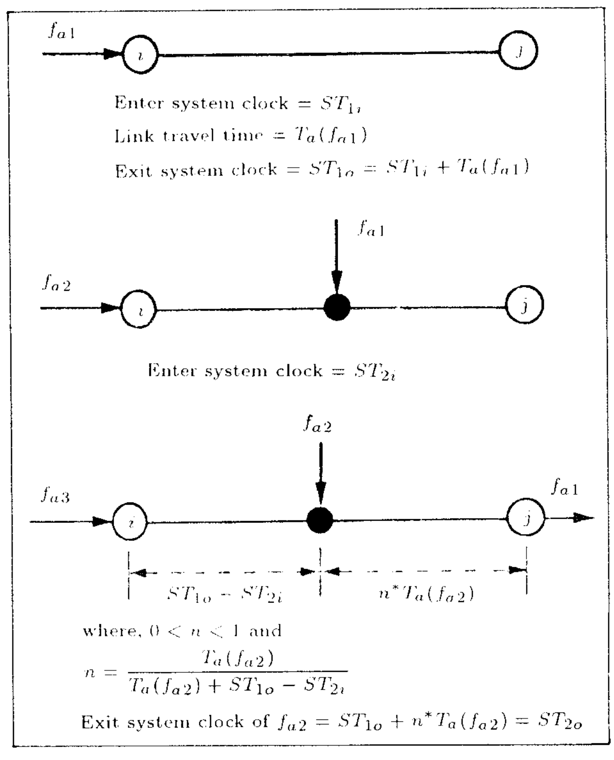


Figure 4. Schematic diagram of remaining link travel time.

travel time that is already spent on the link of f_{a2} is $ST_{1o} - ST_{2i}$, and the remaining travel time of f_{a2} could be viewed as $n^*T_a(f_{a2})$, where $0 < n < 1$. Based on the assumptions indicated in the previous section, the remaining travel time from current position to the closest real node j can be approximately evaluated by the following proportional principle:

$$n = \frac{T_a(f_{a2})}{T_a(f_{a2}) + (ST_{1o} - ST_{2i})} .$$

In this approximation, parameter n approaches zero if $ST_{1o} - ST_{2i}$ is much larger than $T_a(f_{a2})$ which indicates that flow f_{a1} exits much later than when flow f_{a2} entered the link and implies that these two flows are very close. On the other hand, n approaches 1 if $ST_{1o} - ST_{2i}$ is much less than $T_a(f_{a2})$, which indicates flows f_{a1} and f_{a2} enter/exit at almost the same time, thus, the remaining link travel time of flow f_{a2} will not be significantly influenced by f_{a1}. The total link travel time of flow f_{a2} can be estimated as follows:

$$ST_{1o} - ST_{2i} + T_a(f_{a2}) * \frac{T_a(f_{a2})}{T_a(f_{a2}) + (ST_{1o} - ST_{2i})} ,$$

and the exit system clock of f_{a2} can be expressed as:

$$\begin{aligned} ST_{2o} &= ST_{2i} + \textit{total link travel time} \\ &= ST_{1o} + T_a(f_{a2}) * \frac{T_a(f_{a2})}{T_a(f_{a2}) + ST_{1o} - ST_{2i}} . \end{aligned}$$

Summarizing the above explanation, a general form of total travel time of flows on a link can be estimated and is shown below. Flows $f_1, f_2, \cdots f_{n-1}, f_n$ are on a link a where f_1 is in front of f_2 and so on. The enter clock and exit clock values of f_n are ST_{ni} and ST_{no}, respectively. Therefore, ST_{no} can be expressed as:

$$\begin{aligned} ST_{no} &= ST_{(n-1)o} + T_a(f_n) \\ &* \frac{T_a(f_n)}{T_a(f_n) + (ST_{(n-1)o} - ST_{ni})} , \end{aligned}$$

where $ST_{(n-1)o}$ and T_a denote the exit clock value of flow f_{n-1} and link performance function, respectively. The link travel time of next iteration, therefore, is related to the flow being assigned and to the exit time of the previous flow. According to this approximation, the proposed algorithm can assign flows on the network and avoid over/under estimation of the link flows.

Next Earliest Event Problem

An event is defined as any flow that arrives to one of the real nodes. Once the flows have been assigned to any link on the network, there is no way to change their routes until they arrive at the next real node on their path. When the flows arrive at a real node, the algorithm evaluates the network conditions and reassigns the flow. The original routes may be changed when network conditions are changed. To determine the next earliest event on the overall network, it is crucial to check the "updated" link travel times for all the flows already on the network and to find a minimum value. The associated flow of this minimum value is going to reach its next real node prior to that of other flows. Once this minimum value and its corresponding flow is found, the flow will be reassigned based on the network configuration, link performance functions, its original O-D demands and the existing link flows. In the meantime, the system clock will be pushed forward to next period by adding the current system clock value and this minimum value together. The entire procedure, then, operates on and on until all the link flows and O-D demands become zero.

Based on this time interval finding process, no flow can arrive at more than one real node during any time interval. The time interval, therefore, is a variable depending on the minimum value of the "updated" link travel times over the entire network. In addition to the time interval, the system clock is used in the algorithm to monitor the position of each flow on the links and to record the elapsed time of passing events.

An Algorithm for Real Time User Equilibrium Assignment

Now, an algorithm is stated for real-time user equilibrium assignment problem. The inputs of the algorithm are the specification of the original network $G(N, A)$, the link performance functions $T_a(f_a)$ for all $a \in A$ and the O-D demand Matrices. Based on the dynamic time-space network model and the procedures outlined above, the algorithm is as follows:

Step 0. Initialization.

1. Apply disaggregate Frank-Wolfe method to find the link flow pattern over the entire network for the first time period.
2. Update the link flow pattern based on the time interval assumption that the links which are directly connected to the origin nodes keep their assigned flows, and the rest of the links will have zero flow.
3. Set the system clock equal to zero.

Step 1. Stopping rule.

Check if all the links and O-D demands are zero. If not, go to Step 2. Otherwise, stop.

Step 2. Time interval finding.

1. Compare all the link travel times and find the minimum value, T_{min}, associated with the assigned flow f_{Tmin}. The current interval is T_{min}.
2. Set system clock equal to the previous system clock plus T_{min}.
3. Update the origin node of the flow constituting f_{Tmin}.

Step 3. Reassignment process.

1. Check if any external flow rates arrive during the current time interval.
2. Apply disaggregate Frank-Wolfe method to reassign the flow f_{Tmin} and the external flow rates, if any, together.
3. Go to Step 1.

The flow chart of the proposed algorithm is shown in Figure 5.

NUMERICAL EXAMPLES

The algorithm presented previously was tested on two test networks and the results are shown in the following sections. The first network has seven nodes and ten links. This network has a single flow rate, and is used to demonstrate the flow pattern and path travel time difference between the static and the real-time user equilibrium algorithms. The second network has the identical network configuration as the first one, but its O-D demands are varied over time. These flow rates are created by a random number generator.

Test Network 1

The test network 1 has seven nodes which are connected by ten directional links. Nodes 1 and 2 are origins and nodes 6 and 7 are destinations. The O-D demand is constant and unique for every O-D pair and is 40, 30, 20, 70 for O-D pairs 1-6, 2-6, 1-7 and 2-7, respectively. The link performance function of every link is assumed to be of the form $A + BX^4$, where A and B are the link parameters and X is

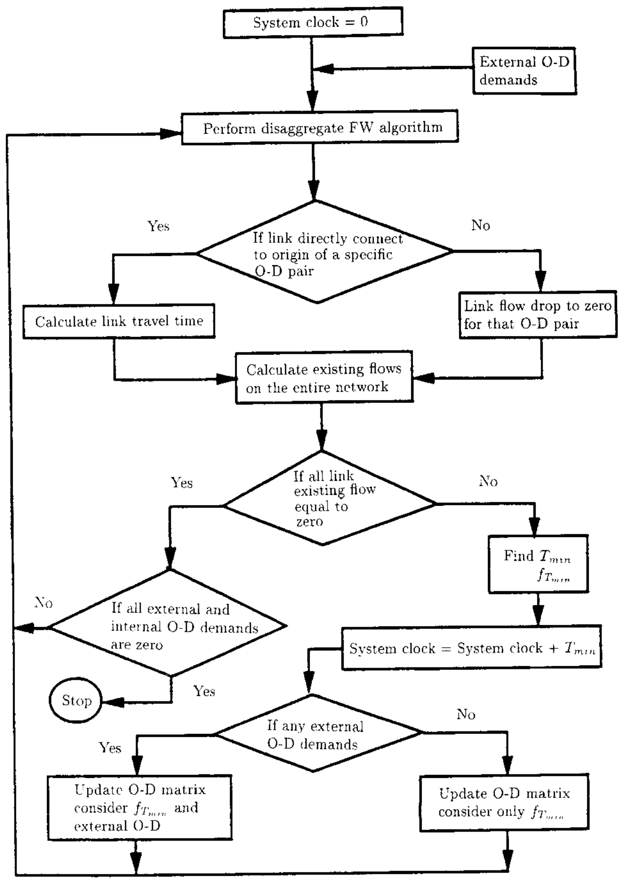
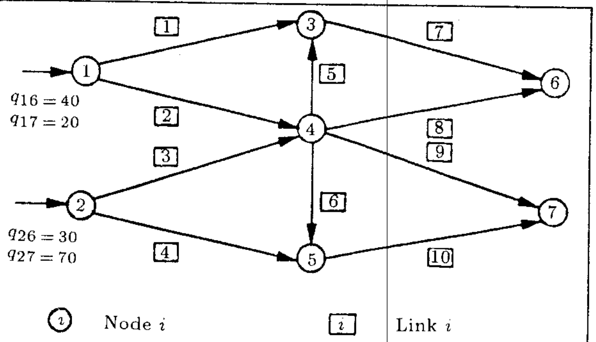


Figure 5. The flow chart of the proposed assignment algorithm.



The general form of link performance function is $A + B^{*}X^{**}4$ where A, B are parameters of each link, X denotes the assigned flow.

Link	A	B
1	5.96	0.23000E-5
2	4.34	0.17000E-6
3	5.17	0.12409E-6
4	8.04	0.19296E-5
5	1.18	0.15504E-6
6	1.08	0.11304E-6
7	9.61	0.10400E-5
8	10.52	0.52500E-5
9	7.47	0.43200E-5
10	12.49	0.21500E-5

O \ D	6	7
1	40	20
2	30	70

Figure 6. Test network 1 and related information.

the flow on the link. All of this information associated with the network itself is shown in Figure 6.

Two convergence criteria are considered in the Frank-Wolfe algorithm. These criteria are the link flow change ratio and the number of Frank-Wolfe algorithm iterations. The link flow change ratio is defined as the value of the current link flow minus the previous link flow divided by the current link flow. If this ratio is less than or equal to 0.001, the algorithm stops. On the other hand, if the number of iterations reaches 2000, the assignment procedure will be stopped. These high convergence criteria are selected to obtain more accurate flow patterns due to the special configuration of this network. A tighter convergence criterion ensures that the link flows converge at higher accuracy. At the same time, the computation time also increases due to this more stringent requirement. In this relatively small network, the difference in computation time because of tighter convergence criteria is insignificant.

In this test, the Frank-Wolfe method is applied to solve the static user equilibrium assignment problem. The results of the link flows, link travel times and path travel times are shown in Figure 1. Examining the path travel time for every O-D pair, the values of each path travel time clearly indicate that the link flow pattern obtained from Frank-Wolfe method are the user equilibrium flows.

The proposed real-time assignment algorithm is now applied to the same network with indentical O-D demands. The problem is solved by fourteen successive user equilibrium procedures till link flows approach zero on the entire network. The system clock is equal to zero at the beginning of the assignment and becomes 29.438 at the end. In stage 1 the external O-D demands are assigned to the network based on the static user equilibrium algorithm. Because the time interval of the next assignment is selected according to the minimum link travel time, only the links which are connected directly to the origins have flows and the other link flows are zero. Hence, links 1, 2, 3 and 4 have their original assigned flow, and flows on the rest of the links are zero.

In stage 2, flow on link 2 (42.721), which includes flows from node 1 to node 6 (22.721) and flows from node 1 to node 7 (20.000), arrives at its closest real node 4, prior to the other flows reaching any real node, due to its shortest travel time. In this node, flow on link 2 is redistributed according to the present network situations. This implies that the original selected shortest route may be changed if network environment is changed. This characteristic provides dynamic capability for any flow that intends to change its route at any real nodes to do so. Flow on link 2, then, is reassigned based on static user equilibrium method, its original O-D demands, existing flow on the network and the volume-delay function of each link. This flow is separated into three parts, which are 7.550, 15.171 and 20.000 on links 5, 8 and 9, respectively. After that, the proposed real-time algorithm explores the links with flow, which are six links in this time, and determines the minimum time that the flows need to reach the next closest real nodes. Flow on link 5 (7.550) becomes the next flow to be reassigned and that is shown in stage 3.

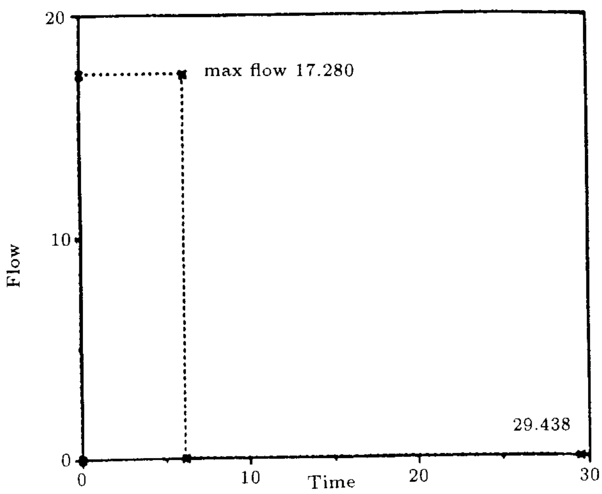


Figure 7a. Link 1 flow-time diagram of test network 1.

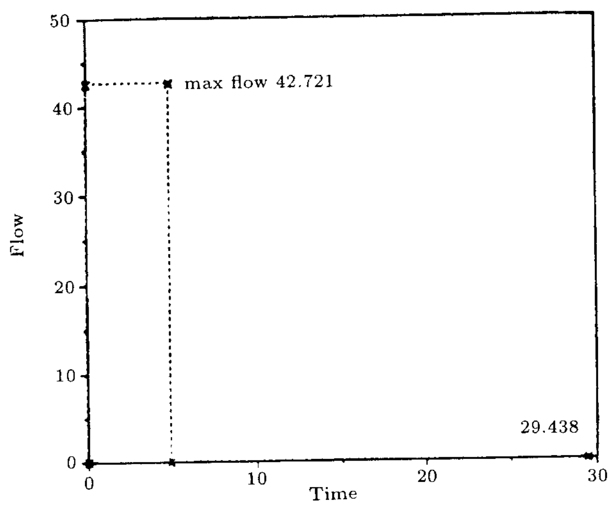


Figure 7b. Flow-time diagram for link 2 of test network 1.

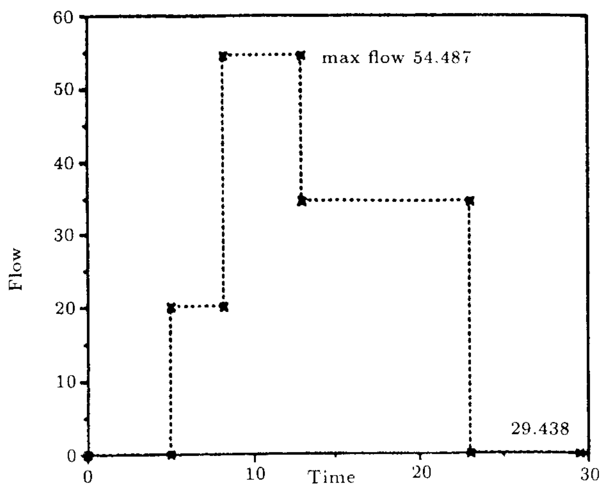


Figure 8a. Link 9 flow-time diagram of test network 1.

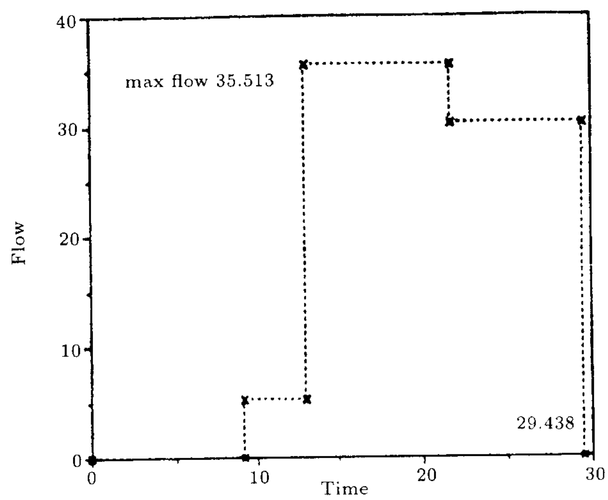


Figure 8b. Flow-time diagram for link 10 of test network 1.

The procedure goes on until all the link flows on the entire network approach zero. At this stage, all of the O-D flows have arrived at their destinations. In this assignment procedure, the link flows are varied over time. The pattern of traffic flow on a link is represented in the form of a histogram which relates the flow to time. Some of these diagrams are shown in Figures 7 and 8. Consider the flow-time diagram of the links 1, 2, 3 and 4. The diagrams have similar shapes which display that all of these links have different amounts of flows in the beginning period of the planning horizon and zero flow during the rest of the time. This is because these four links are directly connected to the origin nodes and a fixed O-D demand is used which enters the network only once at the beginning of the planning horizon. The flow patterns of the rest of the links are varied due to the different level of flows entering or exiting the links. The flow pattern of static method and the maximum flow traversed a link during a specific time interval of the proposed assignment method is shown in Table 1. From this table, it is clear that path flows are essentially different between these two methods except for those links that are directly connected to the origin

Table 1. Comparison of link flows and path travel times between the static and proposed algorithm.

Link	Link Flow (Static)	Maximum Traversed Flow (Proposed Method)
1	17.280	17.280
2	42.721	42.721
3	69.761	69.761
4	30.239	30.239
5	24.231	7.550
6	15.857	5.274
7	41.510	24.830
8	28.489	45.170
9	43.904	54.487
10	46.096	35.513

O-D	Path	Content	Path Flow	Travel Time	Travel Time (Static)
1-6	1	Link 1+7	17.280	20.594	18.863
	2	Link 2+8	15.171	15.704	18.884
	3	Link 2+5+7	7.550	15.700	18.837
1-7	1	Link 2+9	20.000	13.067	28.427
	2	Link 2+6+10	0.000	- - - -	28.190
2-6	1	Link 3+8	30.000	25.460	22.087
	2	Link 3+5+7	0.000	- - - -	22.040
2-7	1	Link 3+9	34.487	23.016	31.630
	2	Link 4+10	30.239	29.438	31.850
	3	Link 3+6+10	5.274	21.680	31.393

nodes. The time-varying link flow pattern, obtained by proposed method, is more likely and closer to the real world network systems because of its realistic time depending and changeable flows.

The path travel times determined by the proposed assignment and its comparisons with the static method are also tabulated in Table 1. One of the interesting phenomena is the dramatic difference between path travel times for O-D pair 1-7. The estimated time of static user equilibrium for this path is 28.190. However, in the proposed real-time assignment procedure, this path has no flow. This indicates that the proposed assignment algorithm dynamically overlooks the situations of the entire network and changes the assigned flows and routes if necessary. Due to the dynamic nature of the proposed algorithm, it is expected that the new path travel time will be less than or at least equal to that of the static method. This is because the real-time assignment algorithm is always looking for an optimization solution during the overall period, not only at the beginning of the planning period. In other words, the proposed algorithm is more "sensitive" to time than traditional static user equilibrium method. The results of the test generally support this conjecture. An example to illustrate this point is to compare the total vehicle-seconds during the entire planning period of the two algorithms. In this network, the total vehicle-seconds of the static and the dynamic methods are 4040.033 and 3536.050, respectively. This clearly demonstrates the improvement of the proposed dynamic traffic assignment algorithm.

The test network 1 is implemented in an IBM VM 370 environment. The running CPU time of this case is 0.74 seconds.

Test Network 2

The test network 2 has an identical configuration to network 1 but with time-varying O-D demands. The purpose of applying time-varying demand is to show that the algorithm is able to handle such flow rates. The time-varying O-D demands are arbitrarily generated by a random number generator and follow a bell-like shape to imitate the peak-hour traffic demand. The shape of time-varying O-D demand is shown in Figure 9. In this test network, O-D demands are generated continuously in time horizon [0,60]. The O-D demand shape can be divided into 12 time slices in which the upper and lower bounds of flow rates are different. The flow rate at each slice has uniform distribution between its upper and lower bound. Once the O-D demand generating criterion, which is defined as current system clock time minus that of the previous being greater than five time unit, is met, the O-D demand generator checks which slice the current system clock is in and generates new O-D demands based on the upper and lower bound of that slice. Then, bell-shape O-D demands will be generated to feed the test network.

The uniform distribution function of a $U(a,b)$ random variable, where a and b denote lower and upper bound, respectively, is easily inverted by applying the aforementioned random number generator. The generating procedure is described below:

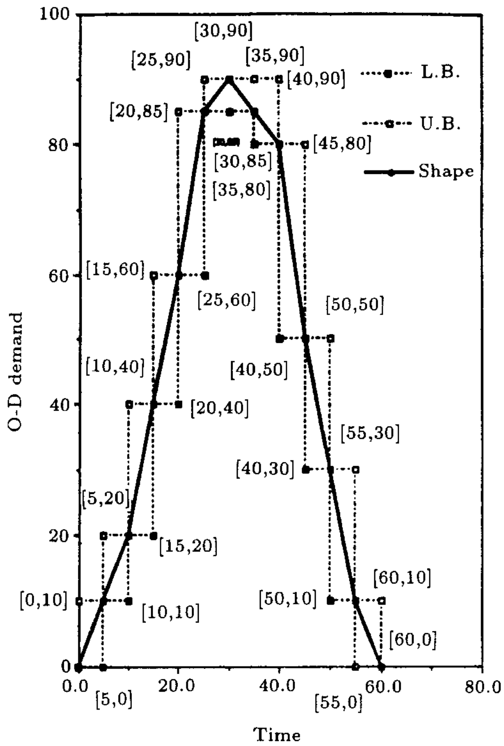


Figure 9. Time varying O/D demand for test network 2.

1. generate a random number u, for $0 \leq u \leq 1$.
2. return the value $x = a + (b - a)u$.

The time-varying demands of the O-D pairs of test network 2, then, can be generated by the above processes which are tabulated in Table 2 and shown in Figures 10a, 10b.

Table 2. Time varying O/D demand of test network 2.

System Clock Time	O-D 1-6	O-D 1-7	O-D 2-6	O-D 2-7
0.000	9.07	2.85	3.69	3.02
5.170	17.64	13.87	15.14	14.96
10.442	24.71	36.82	36.44	21.16
15.742	51.27	41.67	55.02	40.90
21.232	69.96	82.48	72.98	80.12
26.966	87.81	89.76	89.78	87.10
33.484	88.39	89.41	87.91	86.69
38.688	83.83	88.50	89.63	81.20
45.606	33.16	36.50	39.77	38.06
51.172	19.53	20.04	26.23	15.37
59.268	8.32	4.87	0.34	9.99

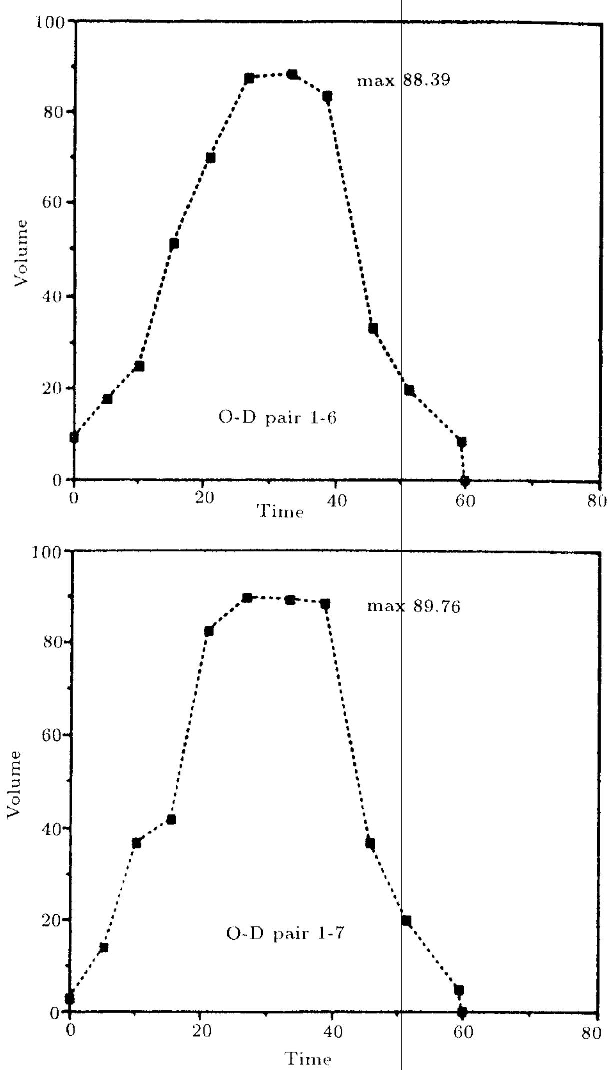


Figure 10a. Flow rates of O/D pairs 1-6 and 1-7 for test network 2.

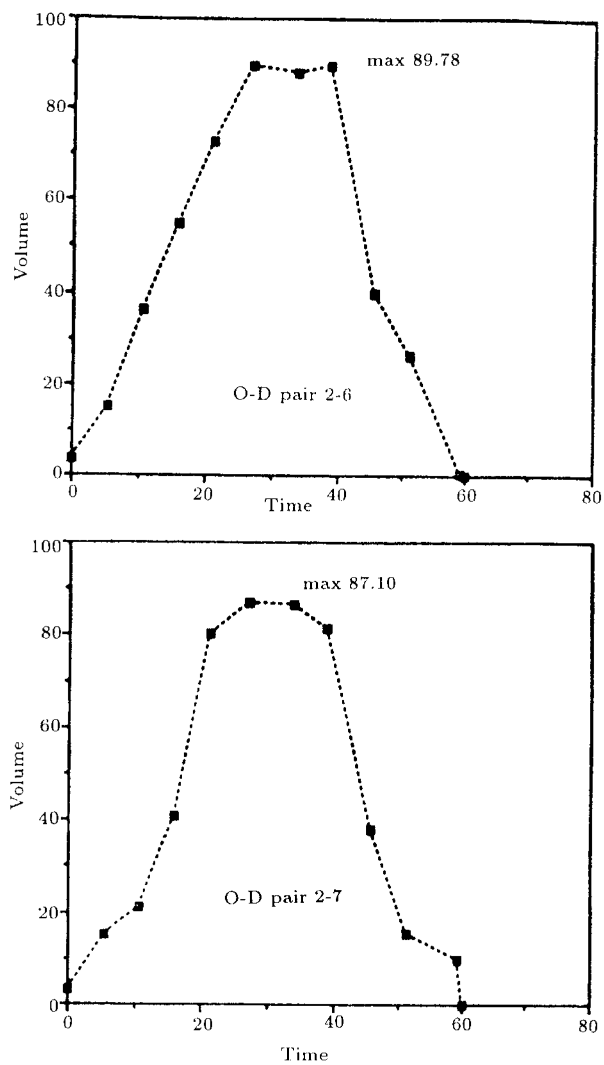


Figure 10b. Flow rates of O/D pairs 2-6 and 2-7 for test network 2.

The proposed assignment algorithm is then applied to solve the assignment task of test network 2 with time-varying demands. Total planning horizon in this case is 146.105 system clock time within 119 consecutive static user equilibrium applications. Some link flows over time diagrams of each link are shown in Figure 11 where link flows are varied with respect to the network traffic conditions during the overall planning horizon. Associated link travel time over planning time horizon diagrams are also shown in Figure 12. The shape of each diagram indicates link travel time variations during the overall planning horizon. Two path flows and their travel times of each O-D pair are tabulated in Tables 3 and 4. In those tables, dynamic routing phenomena can be detected in several paths. As shown in Figure 13 for an O-D pair 1-7, flows are assigned to different paths at different times due to the time depending network conditions.

Flow on link 2(13.847) is reassigned according to user equilibrium requirement at node 4, and traverses link 9 to its destination at 9.677 system clock time. However, another flow on link 2(41.667) is separated into two parts at node 4. One flow with 34.491 units enters link 9 and heads for node 7, its destination, while another with 7.176 units enters link 6 and then passes through link 10 to the same

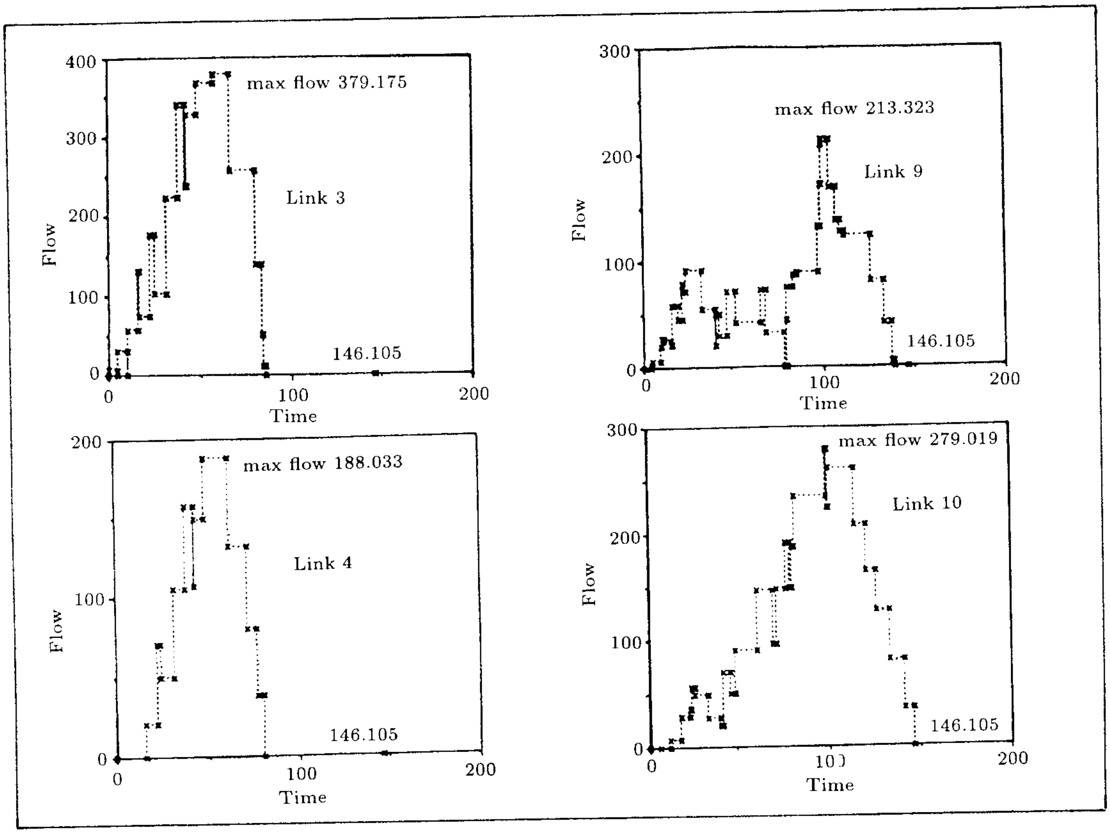


Figure 11. Flow-time diagrams for specific links of test network 2.

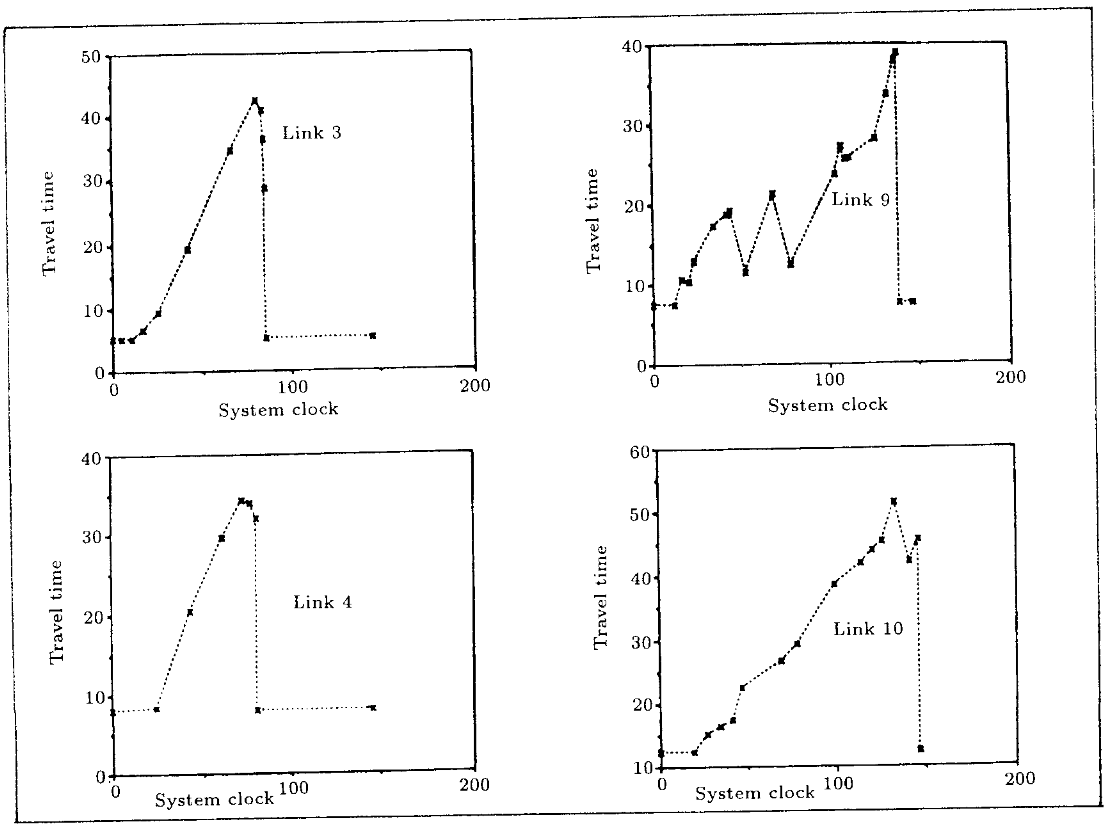


Figure 12. Link travel time vs system clock for test network 2.

destination. Two other flows are reassigned in a similar way at different times. Flows assigned at the same clock time have reasonably close path travel times in this case. Although most of the path travel times are different between O-D pairs, similar paths that enter system clock times have approximately the same path travel times which indicates in this case that the proposed algorithm tries to search the user-optimal assigned flow at each time interval. In addition, there is only one flow assigned to path 2 + 5 + 7 links compared to nine and six assigned flows to path 1 + 7 and path 2 + 8, respectively, between O-D pair 1 − 6. This also indicates that the proposed algorithm explores all the routing and flow distribution options and determines the optimal solution in each time interval. Table 5 demonstrates the approximate equivalence of the path travel times between each O-D pair during different time intervals.

It is difficult to compare the results between static and real-time algorithms because a static method is unable to deal with this

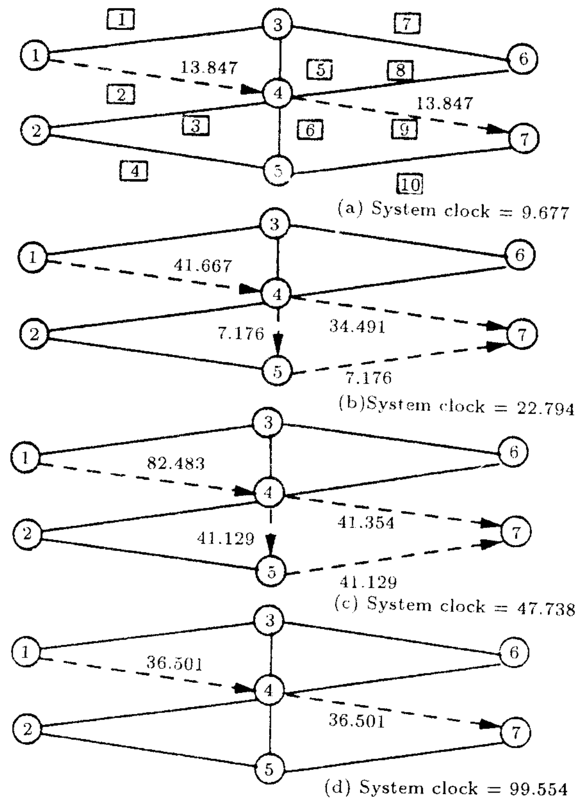


Figure 13. Dynamic routing diagram of O/D pair 1-7 for test network 2.

Table 3. Path flows and travel times for test network 2.

No.	Path Flow	Link 1 In	Link 1 Out	Link 1 Time	Link 7 In	Link 7 Out	Link 7 Time	Travel Time
1	0.036	5.170	11.130	5.960	11.130	22.356	11.226	17.186
2	2.905	10.442	16.473	6.031	16.473	31.510	15.037	21.068
3	30.674	15.768	23.821	8.053	23.821	40.598	16.777	24.830
4	46.974	22.356	39.630	17.274	39.630	57.481	17.851	35.125
5	53.722	31.510	58.611	27.101	58.611	78.036	19.425	46.525
6	57.582	37.849	77.383	39.534	77.383	102.937	25.554	65.088
7	61.484	42.855	97.935	55.080	97.935	133.951	36.016	91.096
8	33.156	48.230	99.242	51.012	99.242	136.542	37.300	88.312
9	4.767	56.824	99.977	43.153	99.977	138.542	38.565	81.718

No.	Path Flow	Link 2 In	Link 2 Out	Link 2 Time	Link 8 In	Link 8 Out	Link 8 Time	Travel Time
1	9.075	0.000	4.343	4.343	4.343	14.899	10.556	14.899
2	17.604	5.170	9.677	4.507	9.677	22.380	12.703	17.210
3	21.803	10.442	16.790	6.348	16.790	33.477	16.687	23.035
4	22.986	22.356	47.738	25.382	47.738	63.641	15.903	41.285
5	34.088	31.510	80.189	48.679	80.189	99.393	19.204	67.883
6	29.192	37.849	97.662	59.813	97.662	121.602	23.940	83.753

Table 4. Path flows and travel times for test network 2.

No.	Path Flow	Link 2			Link 9			Travel Time
		In	Out	Time	In	Out	Time	
1	2.848	0.000	4.343	4.343	4.343	11.814	7.380	11.723
2	13.847	5.170	9.677	4.507	9.677	20.011	10.334	14.841
3	36.822	10.442	16.790	6.348	16.790	34.134	17.344	23.692
4	34.491	15.768	22.794	7.026	22.794	41.538	18.744	25.770
5	41.354	22.356	47.738	25.382	47.738	68.976	21.238	46.620
6	43.944	31.510	80.189	48.679	80.189	103.768	23.579	72.258
7	43.639	37.849	97.662	59.813	97.662	125.798	28.136	87.949
8	39.120	42.855	99.166	56.311	99.166	132.793	33.627	89.938
9	36.501	48.230	99.554	51.324	99.554	137.530	37.976	89.300
10	5.020	56.824	99.954	43.130	99.954	138.770	38.816	81.946

No.	Path Flow	Link 3			Link 8			Travel Time
		In	Out	Time	In	Out	Time	
1	5.943	5.170	10.442	5.272	10.442	27.312	16.870	22.142
2	16.965	10.442	16.978	6.536	16.978	37.849	20.970	27.506
3	23.189	15.768	25.301	9.533	25.301	43.679	18.378	27.911
4	30.155	22.356	41.739	19.383	41.739	56.824	15.085	34.468
5	36.832	31.510	66.165	34.655	66.165	86.347	20.182	54.837
6	35.221	37.849	80.425	42.576	80.425	108.602	28.177	70.753
7	32.525	42.855	83.922	41.067	83.922	115.146	31.224	72.291

Table 5. User equilibrium path travel times for test network 2.

Time Interval	O-D	Path	Flow	Path Travel Time
5	2-6	3+8	30.155	91.159
5	2-6	3+5+7	42.829	87.736
5	2-7	3+9	29.865	164.692
5	2-7	3+6+10	- - - -	- - - -
5	2-7	4+10	50.258	160.269
6	2-6	3+8	36.832	177.120
6	2-6	3+5+7	52.949	183.804
6	2-7	3+9	31.695	226.092
6	2-7	3+6+10	- - - -	- - - -
6	2-7	4+10	55.400	227.231
7	2-6	3+8	35.221	176.454
7	2-6	3+5+7	50.592	171.551
7	2-7	3+9	31.225	212.862
7	2-7	3+6+10	- - - -	- - - -
7	2-7	4+10	52.154	214.150

time-varying demand case. The test network 2 is implemented in the IBM mainframe VM370 environment. Running CPU time for this case is 4.92 sconds.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

This research was devoted to the study of the real-time traffic assignment problem. The assignment algorithms currently used lack the consideration of time variable which causes the methods to be inappropriate in dealing with real world real-time traffic assignment problems. This research has shown that by dividing the planning period into several consecutive time intervals and applying static user equilibrium in an intelligent way in each individual time interval, the dynamic traffic assignment problem can be solved in a relatively easy way. A dynamic time-space network model is created for this purpose. The model is comprised of real and pseudo-nodes, the position of the latter being varied are varied in each individual time interval. The network can be, therefore, viewed as a time dependent structure as well as its O-D demands. The O-D demand which enters the network is viewed as the "external" demand and the flows already on network links are viewed as the "internal" demands. Based on this concept, the network environment during each time interval is in accordance with the initialization requirement of the static user equilibrium. Thus, the created environment allows the convex combination method to be continuously applied in the network. A mathematical program that formulated the dynamic traffic assignment as a discrete time, mixed integer and non-linear programming problem corresponding to user equilibrium on the multiple origins and multiple destination basis is formed based on the time-space flow model. It should be noted, however, that this is not a typical integer program because of the way in which the integer variables in one stage are defined.

Based on the above formulation, a real-time traffic assignment algorithm is developed to distribute a time varying O-D flow matrix over the network and time according to the criteria of user equilibrium.

The operation and performance of the proposed algorithm were examined. Two test networks were employed for this purpose. The first, a small network of seven nodes and ten links associated with a single O-D demand, was chosen to show the operation of the algorithm and to compare the results between static and dynamic methods. The second test network had identical configuration to the first one but with time-varying O-D demands. This network was used to display the application of the proposed algorithm to a small network that has time-varying demands. In the experiments conducted on this network, flows were reassigned on the real nodes in each time interval corresponding to the arriving, exiting and existing flows of each link. Link flow patterns are changed over time which shows the dynamic routing aspects.

In the first network, the link flow patterns are significantly different compared with those of the static user equilibrium assignment method. For example, flows of static method and maximum traversed flows of the proposed algorithm on links 5 and 6 are 24.231, 15.857 and 7.550, 5.274, respectively. These dramatic differences of link flow patterns indicate how essentially different the static and the dynamic methods are. Conclusively, the proposed algorithm is able to reflect the time related assignment problem and to properly assign the time-varying O-D demands with response to the latest network conditions.

One of the difficulties of applying the proposed real time assignment algorithm is the computer's storage memory capacity and its computational speed. Efficient use of the memory and coding of the program would significantly increase the productivity and greatly reduce the computational effort of the algorithm. However, in a real-world network, which may contain hundreds of nodes and thousands of links, using a regular computer to achieve real-time response is impractical. Thus, parallel processing may be the only choice for dealing with real-time network assignment in a real size network. In a parallel processing environment, huge databases can be partitioned into several subdata sets and algorithm can be decomposed into several sub-algorithms and be located at several computing nodes. Then, the data and the algorithm may be synchronously manipulated to reduce the computation and response time. Another possible way for reducing the computation time is the introduction of the concept of neural network in various stages of the assignment. This could have significant potential savings in computation of the path travel times and may under this approach a powerful tool for real-time network assignment be constructed.

Another possible area of research is to investigate the effects of a shorter time interval scale in order to "stabilize" the entering flows. This may be a feasible way to keep network in a relatively stable condition. Once the network is in such a condition, the influences on flows, which are caused by newly entering O-D demands or by network itself, would be minimized. On the other hand, this method would certainly increase the memory storage and computation loading, and its implementation becomes a trade-off judgement.

Anticipating and projecting future demands and determining how they impact traffic assignment in the current time interval is another research approach. Using historical O/D demand data or estimating O/D demands by existing link flows are the potential procedures to deal with the problem. Once the projection of future demands is incorporated in the algorithm, the flow circulation will be significantly decreased and the accuracy and reliability of the algorithm will be increased.

Other possible approaches of research may include incorporating node delay into the model and optimizing intersection (node) signal phasing at the same time, involving elastic O-D demands and concerning link interactions. Because the static user equilibrium is still applicable in each time interval, the techniques originally developed for this purpose can, therefore, be properly utilized into the proposed algorithm. All of these incorporated factors would increase the complexity of the problem and may involve more efforts to resolve the newly-arisen difficulties.

REFERENCES

1. Beckmann, M.J. and McGuire, C.B. and Winsten, C.B. *Studies in the Economics of Transportation*, Yale University Press, New Haven, Conn., USA (1956).

2. LeBlanc, L.J., Morlok, E. and Pierskella, W. "An efficient approach to solving the road network equilibrium traffic assignment problem", *Transportation Research*, **9**, pp 308-318 (1975).

3. Merchant, D.K. and Nemhauser, G.L. "A model and an algorithm for the dynamic traffic assignment problems", *Transportation Science*, **12**, pp 183-199 (1978).

4. Merchant, D.K. and Nemhauser, G.L. "Optimality conditions for a dynamic traffic assignment", *Transportation Science*, **12**, pp 200-207 (1978).

5. Ho, J.K. "A successive linear optimization approach to the dynamic traffic assignment", *Transportation Science*, **14**, pp 295-305 (1980).

6. Carey, M. "A constraint qualification for a dynamic traffic assignment model", *Transportation Science*, **20**, pp 55-58 (1986).

7. Carey, M. "Optimal time-varying flows on congested networks", *Operations Research*, **35**, pp 58-69 (1987).

8. Janson, B.N. "Dynamic traffic assignment for urban road network", *Transportation Research*, B, **25B**(2/3), Washington, DC, USA (1991).

9. Janson, B.N. "A convergent algorithm for dynamic traffic assignment", *70th TRB Conference*, paper # 910735, Washington, DC, USA (1991).

10. Janson, B.N. "Dynamic traffic assignment with scheduled delay", *71st TRB Conference*, paper # 920142, Washington, DC, USA (1992).

11. Zawack, D.J. and Thompson, G.L. "A dynamic space-time network flow model for city traffic congestion", *Transportation Science*, **21**(3), pp 153-162 (1987).

12. Yagar, S. "Dynamic traffic assignment by individual path minimizing & queuing", *Transportation Research*, **5**, pp 179-196 (1971).

13. Van Aerde and Yagar, S. "Dynamic integrated freeway traffic signal network problems and proposed solutions", *Transportation Research* A, **22A**(6), pp 435-443 (1988).

14. Brastow, W.C. "A dynamic traffic assignment model for congested networks with shock waves", Prepared for the Urban Mass Transportation Administration, Stanford University, Distributed by National Technical Information Services, PB-233554 (1973).

15. Luque, F.J. and Friesz T.L. "Dynamic traffic assignment considered as a continuous time optional control problem", Paper presented at the TIMS/ORSA Joint National Meeting, Washington, DC (May 1980).

16. Pontryagin, L.S., Boltyanski, V.A., Gankrelidge, R.V. and Mishenko, E.F. *The Mathematical Theory of Optimal Process*, Wiley, New York, USA (1962).

17. Ran, B., Boyce, D.E. and LeBlanc, L.J. "A new class of instantaneous dynamic user-optimal traffic assignment models", *Operations Research*, **41**(1), pp 192-202 (1993).

18. Boyce, D.E., Ran, B., and LeBlanc, L.J. "Solving an instantaneous dynamic user-optimal traffic assignment models", paper submitted for publication in *Transportation Science* (1991).

19. Ran, B., Boyce, D.E. and LeBlanc, L.J. "Dynamic user-optimal departure time and route choice model: A bilevel optimal-control formulation", paper submitted for publication in the special issue of *Operations Research: Advances in Equilibrium Modeling Analysis and Computation* (1992).

20. Chuang, Y. "An algorithm for real-time user equilibrium traffic assignment", Master's Thesis, University of Maryland at College Park, USA (1992).