Equilibrium Assignment Method for Pointwise Flow-Delay Relationships

A. REGUEROS, J. PRASHKER, AND D. MAHALEL

Most equilibrium traffic assignment models are based on aggregate link performance functions. These flow-delay functions represent a crude abstraction of real dependence of travel time on actual traffic volumes and physical conditions of the transportation network elements. To achieve more realistic assignment results for planning purposes and in the field of intelligent vehicle-highway systems research, several recent works attempt to combine assignment with network simulation. A new equilibrium assignment model that can obtain travel time values from any pointwise volume-delay function is presented. The proposed solution procedure is based on the convex combination method. The proposed assignment procedure is compared with the classic Leblanc's assignment algorithm with fully specified volume-delay functions and with the method of successive averages used in stochastic assignment problems. The proposed method was found to be superior to the MSA procedure due to its faster and more accurate convergence characteristics.

Many transportation planners and researchers have recently faced the need to solve user equilibrium or system optimum network assignment problems without the use of explicitly defined flow-delay functions. Efficient solution methods that exist to handle these assignment problems cannot be used when the flow-delay functions are not explicitly specified as continuous mathematical functions. Typically flow-delay functions such as that developed by the Bureau of Public Roads (BPR) (*1*) and other more sophisticated functions represent crude abstraction of links, intersections, and other network element characteristics. These performance functions reflect the travel impedance associated with the various network elements. The use of these crude aggregate flow-delay functions can easily be justified in long- and medium-range transportation studies where the details of the network's elements cannot be expressed with great accuracy. However, in many recent applications, these aggregate flow-delay functions cannot be used.

When traffic assignment is applied to short-range detailed network planning or as a decision support tool for traffic control strategies, the characteristics of the network elements must be presented with great accuracy. Often simulation models are used to achieve the desired degree of realistic representation of the network element characteristics. The output of the simulation models must be incorporated into an efficient traffic assignment procedure to produce traditional traffic assignment results.

In the field of intelligent vehicle-highway systems (IVHS) research, several recent investigations have identified the impact of various real-time navigation and control strategies on driver behavior and network congestion. Driver behavior in the controlled network environment is usually represented by elaborate simulation models. Based on behavioral assumptions, these models predict how a single driver or a small group of drivers will react to traffic conditions and available route guidance information. The movement of each individual driver is governed by a behavioral simulation model. However, to predict the impact of the proposed control strategies on network flows, the results of the simulation stage must be combined by a mathematical process to achieve internally consistent traffic assignment results. Consistent results, in this context, mean that *a priori* assumptions of the simulation model, for example, regarding travel time considerations, are not violated due to congestion effects.

At present the simulation results are incorporated into user equilibrium or system optimum assignment procedures in one of two methods. In transportation planning applications, a heuristic method of successive iterations that alternates between a flow-delay curve fitting step and a traditional assignment step is usually performed. A typical example of this approach is used by the Simulation and Assignment of Traffic to Urban Road Networks software package (SATURN) (*2*). This software will be discussed further later. In other applications, especially in IVHS research, a method of successive averages (MSA) first suggested by Sheffi (*3*) is frequently used. The two solution methods mentioned above are not completely satisfactory. A more efficient assignment method that can incorporate simulation results into an assignment procedure is still needed.

In the framework of this work, a traffic assignment model is developed that can use flow-delay functions whose explicit mathematical formulation is unknown. This paper proposes a method that can be applied to solve network flow problems. Based on empirical investigation, this solution will be valid as long as the flow-delay curves are nondecreasing when traffic flow increases. The flow-delay function can be a numeric pointwise function or a set of simulation-generated values. Although the proposed solution procedure is stated as a heuristic procedure (the proof of this method is being developed) the presentation follows the Frank-Wolfe method as applied by Leblanc (*4*) to solve for user equilibrium in networks.

ASSIGNMENT AND FLOW-DELAY FUNCTIONS

In traditional transportation planning applications, all the supply characteristics of a given network topology are supposed to be captured by the flow-delay functions. When a complex

Department of Civil Engineering and Transportation Research Institute TECHNION, Israel.

phenomenon like traffic flow is represented by a relatively simple function with a very limited number of explanatory variables, this representation must be crude and aggregate. Therefore, existing flow-delay functions can only coarsely approximate real traffic flow relationships. Yet, due to mathematical convenience and computer efficiency, those primitive flow-delay functions are widely used.

When flow-delay functions are used in an assignment procedure, they must possess some mathematical properties to achieve unique convergence of the assignment procedures. To achieve unique solution of user equilibrium or system optimum assignment under steady-state deterministic conditions, volume-delay curves must satisfy the following properties:

- The functions should be monotone and nondecreasing.
- The functions should be continuous and differentiable.
- The functions should be defined for all positive values.

This means that the function must exist in the region where volume exceeds capacity.

The last property is necessary because in typical assignment applications, inherently nonsteady-state problems are solved as if steady-state conditions prevail. Thus, temporal delays on network elements that experience demand higher than capacity are implicitly accounted for by the oversaturated region of volume-delay functions.

Many flow-delay functions have been developed for use in traffic assignment problems. Most of these functions define the characteristics of links or approaches to signalized intersections. Ortuzar (*5*) reviewed some of these flow-delay curves. Link functions were developed by the early Detroit transportation study and by Davidson (*6*). The function that is most commonly used was developed by BPR (*1*) and is defined as follows:

$$t = t_0\left[1 + \alpha\left(\frac{X}{C}\right)^{\beta}\right] \tag{1}$$

where

t = travel time,
t_0 = free-flow travel time,
X = flow,
C = link's capacity, and
α and β = calibration parameters.

The most commonly used functions to evaluate flow-delay characteristics on signalized intersection approaches are the classic Webster delay function (*7*) and a function developed by Akcelik (*8*). Davidson's link model and Webster's intersection model are not defined when flow exceeds capacity. Travel-time functions of these two models are asymptotic functions, approaching infinite travel times when flow approaches capacity.

Many transportation applications require more realistic assignment results than can be obtained with the existing flow-delay functions. Delay models must be improved and expanded to handle network elements, such as nonsignalized intersections and weaving and merging sections on freeways. To overcome the limitations of coarse and aggregate delay functions, some assignment models use network simulation procedures to evaluate delays. At present, a common characteristic of these assignment-simulation models is an iterative loop between a curve-fitting phase of flow-delay functions based on results generated by the simulation run and a traditional assignment phase. The curve-fitting phase is quite complex, requiring substantial computer time, memory, and storage space to generate the estimated flow-delay curves. A set of flow-delay functions generated in one simulation phase are used in a traffic assignment procedure carried to convergence. Based on the assignment results, successive iterations of the curve-fitting procedure are performed until the process converges (it is hoped). The problem with this process is that typically no *a priori* information exists about the real shape of the flow-delay curves, and there is no assurance that the chosen function used in the curve-fitting phase actually represents the real network element behavior. Futhermore, there is no assurance that the chosen functional form will lead the assignment procedure to converge to the correct solution. Two well-documented applications of this process are SATURN (*9*) primarily developed to handle transportation planning problems and a model developed by Stephanedes et al. (*10*) to assist in traffic control issues.

This paper presents a new assignment methodology to integrate simulation with conventional equilibrium assignment. The suggested approach efficiently uses memory and storage resources, but it does not assume a functional form of the flow-delay relations. This method iterates between simulation and assignment steps until the assignment procedure converges.

EXACT PROBLEM FORMULATION

To achieve a complete presentation of the proposed method, the process begins with a concise derivation of the steady-state user equilibrium traffic assignment problem following Leblanc's (*4*) work. Next, the MSA suggested by Sheffi (*3*) to solve stochastic assignment problems is presented. Finally, the new linearization method (LAM) is presented and compared with the MSA method. The comparison is based on the results of the two methods relative to the solution of an exact Leblanc's algorithm.

Current Equilibrium Assignment Practice

Beckman et al. (*11*) formulated the user equilibrium (UE) problem as a convex (nonlinear) objective function under a set of linear constraints. Leblanc (*4*) proposed an efficient algorithm to solve this problem, when flow-delay functions are fully specified, based on the Frank-Wolfe method (*12*). The steady-state UE problem is formulated as follows:

$$\min f(x) = \sum_{ij} \int_0^x t(w)dw \tag{2}$$

$$\text{st: } D(j, s) + \sum_i x_{ij}^s = \sum_k x_{jk}^s$$

$$x_{ij}^s \geq 0 \qquad \forall\, i, s \tag{3}$$

where

$t(w)$ = a flow-delay function,
x_{ij}^s = flow in link ij destined to s, and
$D(j, s)$ = flow originating at node j destined to s.

Given x^1, a feasible flow vector (which satisfies the conservation of flow equation and the nonnegativity of flow constraints), the first-order expansion of $f(x)$ around x^1 can be written as follows:

$$f(y) = f(x^1) + \nabla f[x^1 + \theta(y - x^1)](y - x^1) \quad \text{for } 0 < \theta < 1 \qquad (4)$$

Assuming θ to be equal to 0, and removing all constant terms, Equation 4 may be rewritten as:

$$\min \nabla f(x^1)y \qquad (5)$$

The new linear program (LP) problem consists of the above objective function subject to the set of conservation flow constraints, that is, Equation 3. The solution to this problem yields a vector y^1 that is also a feasible solution to the original nonlinear problem (Equations 2 and 3). The direction $d = y^1 - x^1$ provides a good direction to seek a reduction in the value of the original objective function $-f(x)$ (*13*). A new value of x^2, which lies between x^1 and y^1, is a feasible solution to the original nonlinear program due to the convex set of flow conservation constraints. To minimize f in the direction d^1, the following one-dimensional problem must be solved:

$$\begin{aligned}&\min f(x^1 + \alpha d^1)\\ &\text{st: } 0 \leq \alpha \leq 1\end{aligned} \qquad (6)$$

The optimal step size, α, can be obtained using an interval reduction method. Further investigation of the objective function, Equation 5, reveals that

$$\frac{\partial x_{ij}}{\partial x_{ij}^s} = t(x_{ij}^1) = c_{ij}^1 \qquad (7)$$

Defining c_{ij}^1 as $t(x|_{x=x1})$. The objective function of the LP can be written as

$$\min \sum_s \sum_{ij} c_{ij}^1 y_{ij}^s \qquad (8)$$

The LP presented by Ortuzar and Willumsen (*5*) can be solved by identifying the shortest paths between all origin-destination (O-D) pairs and assigning all flow to those routes. Based on the above derivation, the solution algorithm may be summarized as follows:

1. Perform initialization. Perform an all-or-nothing assignment based on $t_{ij} = t_{ij}(0)$, and produce a flow vector x^1. Set the iteration counter n to 1.
2. Update travel times. Update the link travel times $[t_{ij}^n = t_{ij}(x_{ij}^n) \quad \forall a]$.
3. Perform direction finding. Perform an all-or-nothing assignment with t_{ij}^n. Define the new flow vector as y_{ij}^n.
4. Perform line search. Find the value of α that minimizes the value of the objective function.
5. Go to the next point. Set $x_{ij}^{n+1} = x_{ij}^n + \alpha_n(y_{ij}^n - x_{ij}^n)$.
6. Perform the convergence test. If the convergence criterion is met, stop. Otherwise, go to Step 2.

Formulation of Assignment Problem with Pointwise Flow-Delay Relationships

To develop an assignment methodology relaxing the requirement of mathematically explicit flow-delay functions, assume that a black box capable of producing delay values for given values of flow exists. Let this black box be defined as a flow delay model (FDM) function, which is schematically presented in Figure 1.

When FDM functions are used, it is impossible to evaluate the objective function of the UE assignment problem:

$$\min f(x) = \sum_{ij} \int_0^x \mathrm{FDM}(w)dw \qquad (9)$$

Observe, however, that when applying Leblanc's (*4*) algorithm to solve the problem, this difficulty affects only Step 4 of the algorithm. The line search for the optimal step size (Equation 6) cannot be solved easily using an FDM model. It requires a continuous evaluation of the objective function (Equation 9) to find its minimum. This step cannot be performed because the functions are not specified, and thus their integral cannot be evaluated. Observe, however, that if an FDM function represents an underlying continuous and monotonic nondecreasing function, all other steps can easily be performed. Thus, the step that in every iteration evaluates the term:

$$\min \sum_{ijk} \frac{\partial f(x_{ij}^1)}{\partial x_{ij}^s} y_{ij}^s \qquad (10)$$

can be substituted by the expression:

$$\min \sum_{ijk} \mathrm{FDM}(x_{ij}^1) y_{ij}^s \qquad (11)$$

To obtain the objective of this work, it is clear that an efficient method to define the step size must be developed. This method must find, at each iteration of the algorithm, a new solution vector x^{n+1} that lies between x^n (the old solution) and y^n.

Method of Successive Averages

MSA was first suggested for use in traffic assignment by Sheffi (*3*). This approach is based on stochastic approximation meth-

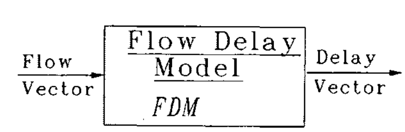

FIGURE 1 Example of a pointwise flow-delay model.

ods. Stochastic approximation is concerned with the convergence of problems that are stochastic in nature, usually based on observations that involve errors. Search techniques that successfully reach optimum despite the stochastic noise were named "stochastic approximation methods" by Robbins and Monroe in 1954 (*14*). The term approximation refers, in this context, to the continual use of past measurements to estimate the approximate position of the solution, and the term stochastic suggests the random character of the function being evaluated.

The Robbins-Monroe procedure places solution point $n + 1$ as a function of the solution of point n according to the following equation:

$$x_{n+1} = x_n + \alpha z(x_n) \qquad (12)$$

where $z(x)$ is a "noisy" function.

The method is based on predetermined step-size series (α) that must satisfy the following two conditions:

$$\sum_{n=1}^{\infty} \alpha_n \rightarrow \infty$$

$$\sum_{n=1}^{\infty} \alpha_n^2 < \infty \qquad (13)$$

In general, any sequence can be used that can be expressed by the equation:

$$\alpha_n = \frac{K_1}{K_2 + n} \qquad (14)$$

where $K_1 > 0$ and $K_2 \geq 0$. One of the simplest step-size sequences that satisfies both conditions is the series:

$$\alpha_n = \frac{1}{n} \qquad (15)$$

Sheffi (*3*) applied this methodology to solve a probabilistic assignment problem. The same approach can be used to solve a deterministic assignment problem when flow-delay relations are expressed as FDM functions. The complete algorithm can be summarized as follows:

1. Perform initialization.
   (a) Run the simulation program with an initial flow vector.
   (b) Perform an all-or-nothing assignment to produce a flow vector x^1.
2. Update travel times. Perform a simulation run with flow vector x^n, generating new travel times (t_{ij}^n).
3. Perform direction finding. Perform an all-or-nothing assignment with t_{ij}^n and define the new flow vector as y_{ij}^n.
4. Go to the next point. Find a point x^{n+1} between x^n and y^n as follows:

$$x^{n+1} = x^n + \frac{1}{n}(y^n - x^n) \qquad (16)$$

Increase the iteration counter n.

5. Perform the convergence test. If the convergence criterion is met, stop. Otherwise, go to Step 2.

The drawbacks and advantages of the algorithm can be attributed to the use of predetermined step sizes. The advantage is the simplicity of the algorithm and its insensitivity to noisy simulation results. One disadvantage is that the algorithm does not directly utilize information obtained with the execution of each simulation step. Thus the convergence is very slow, and appropriate convergence criteria are difficult to define (*15*).

Another disadvantage of MSA may be illustrated with the following example. The MSA algorithm was applied to solve the assignment problem of a network consisting of three links and one O-D pair (see Sheffi (*3*), page 114). Figure 2 shows the objective function, $z(\mathbf{x})$, as a function of the iteration number. If the MSA procedure is ended after a predetermined number of iterations, a poor convergence may be achieved. This can happen because the results of each MSA iteration asymptotically oscillate around the true solution value and do not approach convergence monotonically.

Linear Approximation Method

The proposed method is based on a linear approximation of the real underlying flow-delay function. At each iteration of the Frank-Wolfe algorithm, a new flow-delay point is generated for each network element and a straight line that passes through the last iteration flow-delay point and the present one is calculated. For an errorless FDM function, the straight line will always be a nondecreasing function of volume. The proposed assignment algorithm is composed of a succession of these straight lines used in conjunction with traditional Frank-Wolfe iterations. Theoretically, higher dimension-curve approximation can be developed. The storage requirement in such a case will increase significantly, and it is unclear whether the algorithm's performance will improve. Remember that the higher dimensionality curve must be nondecreasing. Thus, in some cases, the approximation may result in a poor fit. Therefore, the authors chose the simplest of all approximations—a linear one. At each iteration of Frank-Wolfe's algorithm, only two flow-delay points are considered. At each iteration of the algorithm, a straight line is assumed to represent the present flow-delay relationships. During iteration (n), the straight line passes through points x^{n-1} and x^n. Com-

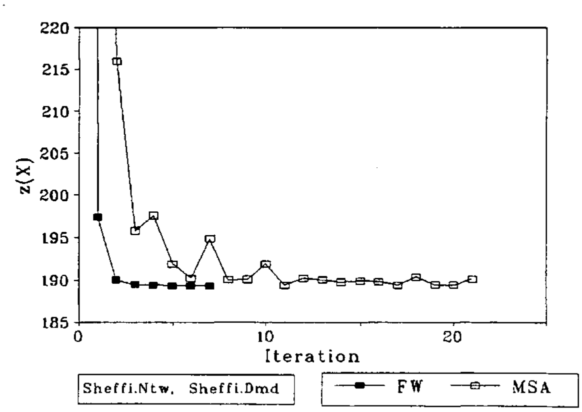

FIGURE 2 Objective function, $z(x)$, as a function of iteration number.

putational efficiency of the LAM approach is achieved by the fact that at any iteration of the algorithm, only one set of flow-delay points for each network element needs to be stored, and travel times on the network elements are evaluated only once.

An example of linear relationship at different iterations is shown in Figure 3. Line *a-b* represents a line that is much different from the real underlying volume-delay functions. Line *c-d* is close to the underlying function in the relevant function interval. If the proposed process actually converges, as empirical evidence indicates, the convergence will occur at a point on the straight line that is a tangent point to the underlying curve. Thus, this convergence point on the straight line is also the convergence point along the "real" underlying volume-delay function.

Mathematically, the formal derivation of the proposed procedure assumes that at each iteration, a linear flow-delay relationship exists according to the following expression:

$$t = \theta_i + \beta_i x_i \tag{17}$$

where β and θ are straight line parameters.

Obviously, if a linear relation exists between flow and delay, then the Frank-Wolfe method can be easily implemented. The temporal (current iteration) objective function is

$$\min z(x) = \sum_{ij} \int_0^x y(w)dw \tag{18}$$

Substituting the linear volume-delay relations into the above equation yields the following expression:

$$\min z(x) \sum_{ij} \left(\theta_{ij}x_{ij} + \frac{\beta_{ij}}{2} x_{ij}^2\right) \tag{19}$$

Thus, the difficulty in algorithm implementation discussed earlier can now be easily resolved. Moreover, when a linear function is used, the optimal step size can be explicitly calculated, eliminating the need for a line-search procedure. This reduces the computational complexity of each iteration of the algorithm. Given two feasible flow vectors, x and y, the line-search step determines the minimum of a function in the interval between the two flow vectors. In case of a linear function, the objective function is convex with respect to x_{ij}. Thus a unique minimum exists in the interval between x and y. The step size can be calculated according to the following expression:

$$\min z[x^n + \alpha(y^n - x^n)] \tag{20}$$

$$st: \quad 0 \leq \alpha \leq 1$$

Defining d^n as the direction between x^n and y^n ($d^n = y^n - x^n$), Equation 20 can be expressed as

$$\min z(x + \alpha d)$$

$$= \min \sum_i \left[\theta_i(x_i + \alpha d_i) + \frac{\beta_i}{2}(x_i + \alpha_i d_i)^2\right] \tag{21}$$

Thus, the optimal step size (α) can be analytically determined according to the following expression:

$$\alpha = -\frac{\sum_{ij} (\theta_{ij}d_{ij} + \beta_{ij}x_{ij}d_{ij})}{\sum_{ij} \beta_{ij}d_{ij}^2} \tag{22}$$

Using the linear function $[z(\cdot)]$ and step size (α), the Frank-Wolfe algorithm can be implemented to solve assignment problems using pointwise flow-delay relationships. At each iteration of the algorithm, a better approximation of the original function is achieved. The proposed assignment algorithm is heuristic in the sense that no formal mathematical proof exists at present. Empirical evidence suggesting convergence to the correct solution will be presented in the next section. Observe that if the iterative process is moving in the right direction toward convergence, then as the process progresses, the difference between the underlying volume-delay function and the succession of straight lines becomes smaller and smaller. This indicates, at least intuitively, that the process should converge to the right solution.

The proposed algorithm can be summarized as follows:

1. Perform initialization.
   (a) Calculate an initial delay vector based on FDM.
   (b) Perform an all-or-nothing assignment to produce a flow vector x^1.
2. Update travel times. Calculate the delay vector based on flow vector x^n, let FDM (x^n) = t^n.
3. Perform linearization. Calculate the linear function $[z(x)]$ based on vectors x^{n-1} and x^n.
4. Perform direction finding. Perform an all-or-nothing assignment with t^n. Define the new flow vector as y_{ij}^n.
5. Go the the next point.
   (a) Calculate the step size according to Equation 29.
   (b) Set $x_{ij}^{n+1} = x_{ij}^n + \alpha(y_{ij}^n - x_{ij}^n)$.
   (c) Increase the iteration counter n.
6. Perform the convergence test. If the convergence criterion is met, stop. Otherwise, go to Step 2.

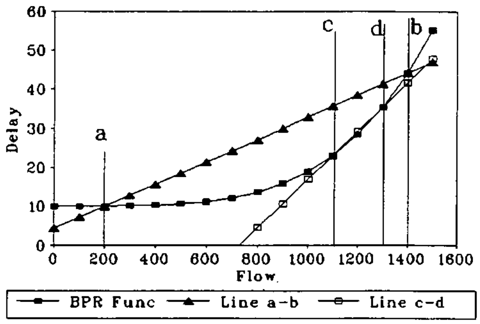

FIGURE 3 Linearized flow-delay relationship.

Examples and Results

To determine the ability of the proposed algorithm (LAM) to provide accurate assignment results, the method was tested on three different networks. The proposed assignment methodology was compared to Sheffi's. Leblanc's implementation of the Frank-Wolfe procedure was used as the yardstick to evaluate the accuracy of the two methods. The proposed algorithm and the MSA procedure were implemented using a BPR function to calculate delays, but it was assumed that the delay values are the result of a pointwise FDM model. The BPR functions were evaluated at discrete points, as if it were impossible to calculate the original objective function integral [$\int t(w)dw$]. For comparison purposes, Leblanc's procedure was implemented using explicitly specified BPR functions. For each experiment, many iterations of the algorithm were performed to ensure convergence.

To examine the sensitivity of the proposed procedure to different congestion conditions, assignments were performed with different underlying volume-delay functions. These flow-delay relationships were based on BPR functions (*1*) (Equation 1) with different α and β values. It should be expected that the lower the congestion, the better the proposed method will perform. At the extreme when no congestion effects exist, that is, travel times are constant, the proposed method will converge after only one iteration, as will the Frank-Wolfe procedure. Several tests were performed to gain insight into the sensitivity of the LAM and MSA procedures to congestion (nonlinearity) effects.

The proposed method was initially applied to the very simple three links network presented by Sheffi (*3*). Figure 4 shows the convergence pattern of the three methods. It can be seen that the proposed method converges asymptotically to the exact solution. For this small example, the performance of the proposed methodology is better than that of the MSA method in two aspects. First, it steadily converges to the exact solution. Second, the number of iterations necessary to achieve an acceptable solution is significantly smaller.

The method was also applied to a 9-node and 16-link grid network. The results obtained by the proposed method were always better than those obtained by the method of successive averages. Finally, the method was applied to two larger networks. First the classic network of Sioux Falls, presented in the original work by Leblanc (*7*) was investigated. It consists of 24 nodes and 76 links. Next, the new algorithm was applied to the network of the classic city of Jerusalem.

For the Sioux Falls network, many UE assignment runs with different BPR volume-delay functions were performed to examine congestion effects of the convergence properties on the proposed methods. The different flow-delay relations were defined by changing α and β values of the BPR function. The higher the value of β, the more sensitive is the function to congestion effects. When β equals 1, a linear relation exists between volume and delay. For comparison purposes, 25 iterations of the LAM algorithm and the MSA method were performed for each volume delay curve. As expected, the proposed method produced better results than the MSA method. After 25 iterations of the algorithm, the proposed method was always closer to the exact solution obtained by Leblanc's algorithm. Figure 5 presents the results of one of the experiments. Table 1 presents assignment results for various α and β combinations of the BPR function parameters. It can be seen that no matter which function was used, the LAM algorithm results are closer to the exact solution than those of the MSA method. Furthermore, the convergence characteristics of the proposed method do not deteriorate significantly when the sensitivity of the network elements to congestion increases. It should be noted that the MSA method is even less sensitive to congestion effects. However, even when $\beta = 5$, the proposed method was significantly closer to the Frank-Wolfe solution than was the MSA method.

The LAM algorithm was applied to a real planning problem in the city of Jerusalem. The network consisted of 639 nodes, 1,621 links, and 9,774 O-D pairs. As before, the results were compared to Frank-Wolfe and MSA algorithms and are presented in Figure 6. Here again, the proposed method outperforms the MSA. Even more surprising, after 25 iterations, the results of the proposed method are extremely close to the Frank-Wolfe solution. Each iteration of the LAM procedure is shorter than that of the Frank-Wolfe algorithm due to its simpler step-size calculations. Typical run times of 25 iterations of the three algorithms on an ISA 486 computer of the Jerusalem network were 1,280 sec for the Frank-Wolfe algorithm, 1,167 sec for the LAM algorithm, and 1,075 for the MSA procedure. Thus, the run time of the LAM procedure was about 10 percent shorter than Frank-Wolfe and about 10 percent longer than MSA.

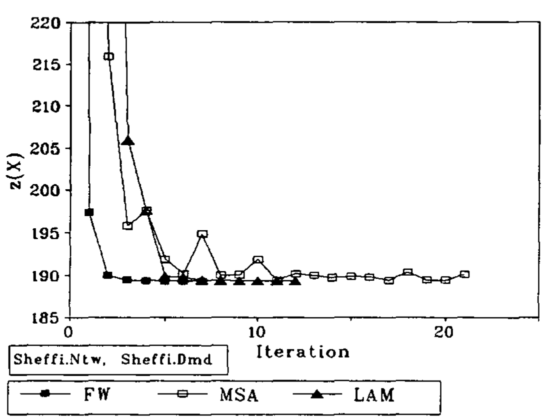

FIGURE 4 Convergence pattern of three link network.

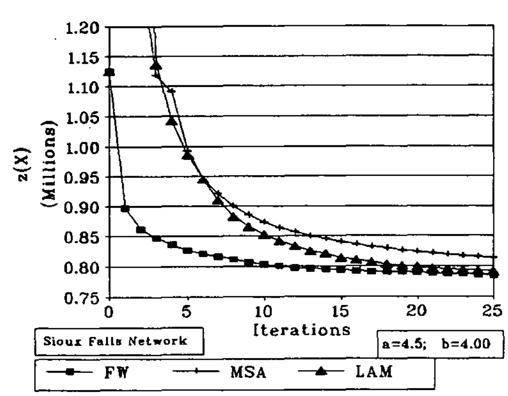

FIGURE 5 Convergence pattern of Sioux Falls network.

TABLE 1 Objective Function Values for Sioux Falls Network

Parameter		Method			Results (percent)	
α	β	FW	MSA	LAM	FW-MSA	FW-LAM
0.15	1	673465.25	673465.06	673465.42	0.0000	0.0000
0.15	2	658047.13	658092.72	658048.79	0.0069	0.0003
0.15	3	652008.50	652094.23	652008.11	0.0131	−0.0001
0.15	4	649040.33	649380.82	649056.72	0.0525	0.0025
0.15	5	647383.66	647816.21	647397.8	0.0668	0.0022
3.00	1	1251805.93	1262714.46	1251836.66	0.8714	0.0025
3.00	2	920306.65	938585.44	920602.65	1.9862	0.0322
3.00	3	797500.07	814918.08	800769.58	2.1841	0.4100
3.00	4	741256.24	756196.98	744504.26	2.0156	0.4382
3.00	5	711110.66	732911.87	713986.46	3.0658	0.4044
4.50	1	1052317.34	1081560.57	1054371.86	2.7789	0.1952
4.50	2	1052317.34	1081560.57	1054371.86	2.7789	0.1952
4.50	3	868625.09	895160.58	873920.02	3.0549	0.6096
4.50	4	784554.39	812435.01	790197.29	3.5537	0.7192
4.50	5	742024.08	769401.7	748640.45	3.5896	0.8917

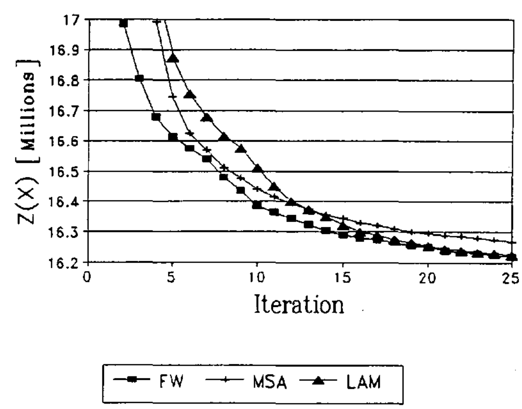

FIGURE 6 Results of Jerusalem network.

CONCLUSIONS

The proposed linear approximation assignment method appears to work well. When an errorless deterministic FDM exists, the proposed method is clearly superior to the MSA method in its convergence properties. An advantage of the proposed method is that it provides an elegant, simple, and computer-storage-efficient iterative procedure to combine traffic assignment with simulation results. Another significant characteristic of the proposed method is that its convergence characteristics are not sensitive to congestion effects.

The proposed method can easily be adapted to situations in which some of the network elements are represented by aggregate flow-delay curves, and the behavior of others is determined by FDM functions. The need to combine two types of volume-delay functions arises when the behavior of some network elements is too complicated to be represented by an aggregate function. When performing microassignment or assignment used to support traffic control decisions, network elements, such as intersection approaches, weaving sections, ramps, and other similar elements, need to be represented in detail. On the other hand, many elements do not need such a fine representation. Furthermore, this method also seems well suited to be applied as a second-stage refined assignment procedure to a solution vector that was generated from aggregate flow-delay functions.

Procedures that perform stochastic assignment are of great interest. The ability of the proposed procedure to perform stochastic assignment was not fully investigated. When the method is applied to solve stochastic assignment problems, the slope of the flow-delay line, in some iterations, may be negative. This violates one of the properties necessary for the convergence of the Frank-Wolfe algorithm. However, due to the stochastic properties of the process, the slope will be negative only during part of the iterations. Thus it may, although not necessarily will, imperil the convergence of the procedure. A probable way to overcome the problem is to assign a zero slope, or to use the slope value of the previous iteration. The probability that the slope will be negative increases as the distance between the two points defining the straight line decreases and as the gradient of the underlying flow-delay curve between the two points approaches zero. Due to the complexity of the convergence process of stochastic assignment, the convergence characteristics of the proposed method when performing stochastic assignment needs further investigation.

ACKNOWLEDGMENTS

This research was supported by a grant from the Israel Ministry of Transport. Special thanks are due to Lyn Long of the University of California, Irvine, for reviewing and editing this paper.

REFERENCES

1. *Traffic Assignment Manual*, Bureau of Public Roads, Urban Planning Division, U.S. Department of Commerce, 1964.
2. Hall, M. D., D. Van Vliet, and L. G. Willumsen. SATURN—A Simulation-Assignment Model for the Evaluation of Traffic Management Schemes. *Traffic Engineering & Control*, 1980, pp. 168–175.
3. Sheffi, Y. *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*. Prentice-Hall, Inc., Englewood Cliffs, N.J., 1985.

4. Leblanc, L. J. *Mathematical Programming Algorithms for Large Scale Network Equilibrium and Network Design Problems*. Ph.D. thesis. Northwestern University, Evanston, Ill., 1973.
5. Ortuzar, J., and L. G. Willumsen. *Modelling Transport*. John Wiley and Sons, Inc., New York, N.Y., 1990.
6. Davidson, K. B. A Flow Travel Time Relationship for Use in Transportation Planning. *Proc., Australian Road Research Board*, Vol. 3, 1966, pp. 183–194.
7. Webster, F. V. *Traffic Signal Settings*. Road Research Laboratory Technical Paper 39. Her Majesty's Stationery Office, 1958.
8. Akcelik, R. The Highway Capacity Manual Delay Formula for Signalized Intersections. *ITE Journal*, Vol. 58, 1988, pp. 23–27.
9. Van Vliet, D. SATURN—A Modern Assignment Model. *Traffic Engineering & Control*, 1982, pp. 578–581.
10. Stephanedes, Y. J., I. Z. Argiropoulos, and P. Michalopoulos. On-Line Traffic Assignment for Optimal Freeway Corridor Control. *Journal of Transportation Engineering*, Vol. 116, No. 6, 1990, pp. 744–755.
11. Beckman, M., C. B. McGuire, and C. B. Winston. *Studies in the Economics of Transportation*. Yale University Press, New Haven, Conn., 1956.
12. Avriel, M. *Nonlinear Programming: Analysis and Methods*. Prentice-Hall, Inc., Englewood Cliffs, N.J., 1976.
13. Zangwill, W. I. *Nonlinear Programming: A Unified Approach*. Prentice-Hall, Inc., Englewood Cliffs, N.J., 1969.
14. Wilde, D. J. *Optimal Seeking Methods*. Prentice-Hall, Inc., Englewood Cliffs, N.J., 1964.
15. Powell, W. B., and Y. Sheffi. The Convergence of Equilibrium Algorithms with Predetermined Step Sizes. *Transportation Science*, Vol. 16, No. 1, 1982, pp. 45–55.

Publication of this paper sponsored by Committee on Transportation Supply Analysis.