UCI-ITS-WP-91-8

Equilibrium Assignment Method for Pointwise Flow Delay Relationships

UCI-ITS-WP-91-8

Joseph N. Prashker [1]
David Mahalal [2]
Andre Regueros [2]

[1] Visiting Researcher, Institute of Transportation Studies
University of California, Irvine, and Department of Civil Engineering and
Transportation Research Institute, Technion - Israel Institute of Technology

[2] Department of Civil Engineering and
Transportation Research Institute, Technion - Israel Institute of Technology

August1991

Institute of Transportation Studies
University of California, Irvine
Irvine, CA 92697-3600, U.S.A.
http://www.its.uci.edu

Introduction

Most of the equilibrium traffic assignment models used nowadays, 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. These link performance functions reflect the travel impedance associated with the links and intersections. In many applications, especial those which are concerned with detailed microscopic traffic analysis, the performance of these simplified flow-delay relationship might be too crude and thus unsatisfactory. When such analysis is desired, detailed flow-delay models, or simulation models, have to be used. Furthermore in many investigations different levels of detail are necessary for various components of the network. The flow-delay characteristics of some network elements can be represented by crude aggregate relations while other elements need to be represented in great detail and accuracy. When some, or all, of the network elements are not represented by mathematically defined flow-delay function it becomes very difficult to solve for user equilibrium in a transportation network. Similar difficulties might arise in the investigation of system optimum of transportation, communication or other networks.

In the framework of this work, a traffic assignment model is developed that can be based on functions, whose exact mathematical form is not known. The proposed solution method applies to steady state network flow problems. This solution will be valid as long as the flow-delay curve is non deceasing when traffic flow increases. The flow-delay function can be numeric pointwise function or a set of simulation-generated values. The empirical analysis and derivation of the proposed solution methods follows the user equilibrium, traffic assignment model, developed by Leblanc [7].

Link Performance Functions

When solving for equilibrium assignment, one has to pay attention to how the travel time is related to traffic volume and to other characteristic. Most of these flow-delay models are based on crude and aggregate relationships, and represent, therefore, only in an approximate and coarse manner real traffic flow conditions.

The equilibrium assignment model requires that these flow-delay curves satisfy a number of properties:

* The function should be monotone and non-decreasing.
* The function should be continuous and differentiable.
* The function must be defined for oversaturated regions (during the assignment process, some links will be loaded with more traffic than its capacity).

The last property is necessary when solving transportation networks, because inherently non steady state problems are solved as if steady state conditions prevail. Thus, temporal delays on network elements, which experience demand higher than capacity, are implicitly accounted for by the oversaturated region of the flow-delay function. A number of authors have suggested functional forms for flow-delay relationships. Ortuzar [8] review some of these flow-delay curves:

1. The Detroit Study:

$$t = t_0 \exp\left(\frac{x}{C}\right) \tag{1}$$

   where t is the travel time, t_0 is the free flow travel time, x is the flow, and C is the link's capacity.

2. The Bureau of Public Roads in the USA proposed the most common function:

$$t = t_0\left[1+\alpha\left(\frac{x}{C}\right)^{\beta}\right] \tag{2}$$

where α and ß are parameters for calibration

3. A function that is asymptotic to a capacity flow was proposed by Davidson [5] based on queuing theory considerations:

$$t = t_0\left[1+J\frac{x}{c-x}\right] \tag{3}$$

where J is a parameter of the model.

4. When dealing with signalized networks other functions have to be employed. Almost any model that relates delay caused to the traffic flow, to traffic signal parameters (cycle length, effective green time, saturation flow) can be employed. One of the most frequently used delay models is due to Webster [13]:

$$d = \frac{c(1-\lambda)^2}{2(1-\lambda y)} + \frac{y^2}{2x(1-y)} - 0.65\left(\frac{c}{x^2}\right)^{1/3} y^{2+5\lambda} \tag{4}$$

where
- d average delay per vehicle
- c cycle time
- λ proportion of the cycle which is effectively green (g/c)
- x traffic flow
- s saturation flow
- y the degree of saturation.

Webster's model does not apply in oversaturated conditions when y=>1.

5. Akcelik [2] developed an improved traffic delay model for signalized intersections. This model is valid for udersaturated as well as oversaturated conditions: where notation is as above with the following additions:

$$d=\frac{0.5c(1-\lambda)^2}{1-\lambda y}+900Ty^n\left[(y+1)+\sqrt{(y-1)^2+\frac{m(y-y_0)}{QT}}\right] \quad \textbf{(5)}$$

T	flow period in hours
Q	capacity in vehicles per hour
m, n	calibration parameters
y_0	the degree of saturation below which the second term in equation (5) is zero.

Some of the models shown above are not defined when flow exceeds capacity. Davidson's model and Webster's (Equations (3) and (4)) do not work in the oversaturation region. These two models are asymptotic functions, meaning that they generate infinite travel time, when flow is equal or greater than capacity.

It should also be noted that all of the above models include only a limited number of variables and are therefore not realistic enough for congested urban areas. In order to obtain more realistic assignments, the delay models involved must be improved, and expanded to handle many network elements such as nonsignalized intersections, weaving and merging sections on freeways etc. In order to overcome the disadvantage of using an incomplete set of empiric and aggregate delay functions, some assignment models use fine scale simulation of the delays. These delays are then used by the assignment model. At present, a common characteristic of such models is an iterative loop between a curve fitting phase of flow-delay functions based on simulation results and a traditional assignment phase. The curve fitting phase is quite complex, requiring a lot of computer time, memory and storage space, to generate the estimated flow-delay curves. Those curves are used in a complete traffic assignment procedure. Based on the assignment results a new iteration of the curve fitting procedure is performed and so on until the process hopefully converges. The problem with this process is that in many cases we have no a priori information about the shape of the flow-delay curve, and have no assurance that the chosen form represents actual behavior, and will converge to the correct solution.

The present paper presents a new assignment methodology which integrates simulation with conventional equilibrium assignment. This method overcomes some of the drawbacks of the existing methods. It does not assume any functional form of the flow-delay relations, and uses efficiently memory and storage resources. The proposed method iterates between simulation and assignment steps however, convergence of the assignment procedure is reached only once in the proposed process.

Simulation and Assignment

A number of assignment models that are based on flow-delay values obtained from simulation programs have been developed. Their common characteristic is an iterative loop between the simulation and curve fitting phase on one hand and a whole converged assignment phase on the other. This iterative process is repeated until some convergence criterion is satisfied. It is worth noting that no convergence can be warranted by means of such an algorithm. An other disadvantage of these algorithms is that they repeatedly perform to completion a number of equilibrium assignment procedures. A brief description of two of such models will be presented in the following paragraphs.

The SATURN Model

SATURN [6] (Simulation and Assignment of Traffic to Urban Road Networks) is a computer model developed at the Institute for Transport Studies, University of Leeds, for the analysis and evaluation of traffic management schemes.

SATURN uses two sub-models in order to achieve "realistic" assignments [12]. The first one is a TRANSYT type simulation model based on the use of cyclic flow profiles to represent the movements of platoons of vehicles over a network. It needs information about the flow on each link of the network to estimate capacity, queues and delays. Therefore, an assignment model is required to load a trip matrix onto the network and obtain an estimate of these flows. This is achieved through an separate assignment model. The link between these two models is through the flow delay curves as shown in Figure 1.

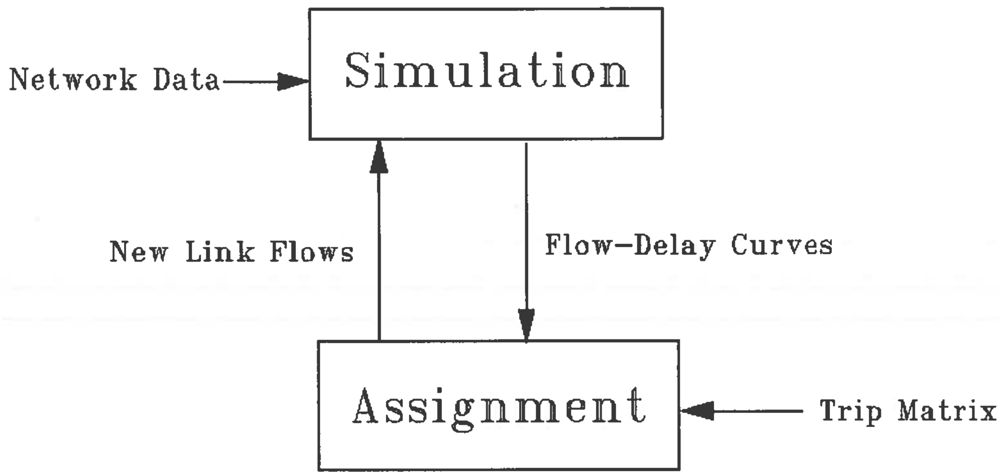

Figure 1: The Simulation and Assignment Phases of SATURN

The objective of the simulation phase is to generate flow-delay relationships from a given pattern of traffic flows in a network. These flow-delay curves are obtained by calculating the delays for each movement at zero flow, current flow (results of the last assignment procedure) and capacity with all other flows (i.e. opposing traffic) fixed. With these three points a flow-delay curve, that take the form of a polynomial, is fitted:

$$d(x) = \begin{cases} d_0 + ax^n & x < C \\ d(C) + \dfrac{T(x-C)}{2C} & \text{Otherwise} \end{cases} \qquad \textbf{(6)}$$

where: $d(x)$ average intersection delay experienced by traffic flow x

C turn capacity

d_0 delay at 0 flow

a, n parameters

T duration of the simulation period

The iterative process continues until the turning movements reach reasonable stable values (i.e. the flow patterns are similar in two consecutive iterations). It must be noted that ultimate convergence to stable values is difficult [6].

Other Models

Stephanedes [11] developed a simulation-assignment model based on an iterative feedback loop between an assignment and a simulation phase. The assignment phase distributes trips to the network and the simulation phase provides detailed information about the network performance given its geometric and operational characteristics. Like in the SATURN model, the loop terminates when the travel times of the links between two successive iterations reach reasonably stable values.

The objective of the simulation phase is to provide detailed information about link travel times resulting from a given traffic-flow pattern. This information includes a significant number of <flow>, <delay> points used in a statistical estimation of volume-delay curves. These fitted delay curves are used then in the assignment phase to distribute flow over the network.

Exact Problem Formulation

For sake of completeness of the presentation we start with a concise derivation of the steady state user equilibrium traffic assignment problem following Leblanc's [7] work. Next the method of successive averages - MSA suggested by Sheffi [10] for the solution of stochastic assignment is presented. Finally a new linearization method is presented and compared to the MSA method.

Current Equilibrium Assignment Practice

Beckman *et al.* formulated the user equilibrium problem (UE) as a convex (nonlinear) objective function and a set of linear constraints. LeBlanc [7] proposed an algorithm to solve this problem when the flow-delay functions are fully specified based on the Frank-Wolfe method (see Avriel [3]). The steady state UE problem is formulated as follows:

$$\min \quad f(x) = \sum_{ij} \int_0^x t(w)\, dw \tag{7}$$

$$\begin{aligned} \text{st: } D(j,s) + \sum_i x_{ij}^s &= \sum_k x_{jk}^s \\ & \qquad \forall\ i,s \\ x_{ij}^s &\geq 0 \end{aligned} \tag{8}$$

Where $t(w)$ is a flow-delay function, $X_{ij}{}^s$ is the flow on link {ij} to destination - s, and $D(j,s)$ is flow originating at node j destined to s. Given $\mathbf{x}^1$ a feasible flow vector (a flow vector that satisfies the conservation of flow equation and the nonnegativity of flow constraints), then a first order expansion of $f(\underline{\mathbf{x}})$ around $\mathbf{x}^1$ can be written as:

$$f(\boldsymbol{y}) = f(\boldsymbol{x}^1) + \nabla f(\boldsymbol{x}^1 + \theta(\boldsymbol{y} - \boldsymbol{x}^1))(\boldsymbol{y} - \boldsymbol{x}^1) \qquad \text{for } 0<\theta<1 \tag{9}$$

A linear approximation to $f(\underline{\mathbf{y}})$ is to let θ equal 0 (this yields a linear function in $\underline{\mathbf{y}}$). Further manipulation of equation (9) and removal of all constant terms yields the following objective function:

$$(LP:) \qquad \min \ \nabla f(\boldsymbol{x}^1)\boldsymbol{y} \tag{10}$$

Solving the above LP problem under a set of conservation flow constrains, equation (7), yields a solution vector $\underline{\mathbf{y}}^1$ which is also a feasible solution to the original non linear problem equations (7) & (8). The direction $\underline{\mathbf{d}}=\underline{\mathbf{y}}^1-\underline{\mathbf{x}}^1$ is a good direction to seek a decreased value of f (see Zangwill [15]).
Since the feasible region (determined by the flow conservation equations) is convex, each point on the line between $\underline{\mathbf{x}}^1$ and $\underline{\mathbf{y}}^1$ is also feasible. So, to minimize f in the direction $\underline{\mathbf{d}}^1$ a one dimensional problem,

$$\begin{aligned} &\min \ f(\boldsymbol{x}^1 + \alpha \boldsymbol{d}^1) \\ &st: \quad 0 \le \alpha \le 1 \end{aligned} \tag{11}$$

has to be solved. The optimal step size, α, can be obtained from any interval reduction method. Further investigation of the LP objective function, equation (10), reveals that:
So that

$$\frac{\partial x_{ij}^1}{\partial x_{ij}^s} = t(x_{ij}^1) \tag{12}$$

Defining c_{ij} as $t(x|_{x=x1})$, the linear program (LP) can be written as:

$$\min \ \sum_s \sum_{ij} c_{ij} y_{ij}^s \tag{13}$$

This program can be minimized by finding the shortest path connecting each OD pair and assigning all the flow to it [7,10].
The algorithm can be summarized as follows:

1. Initialization
   Perform *All Or Nothing assignment* based on $t_{ij}=t_{ij}(0)$. This yields to 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. Direction Finding
   Perform an *All or Nothing assignment* with t_{ij}^n. This yields the auxiliary flow vector y_{ij}^n
4. Line Search
   Find α that solves the linear program (see Equation (11)).
5. Move
   Set $x_{ij}^{n+1}=x_{ij}^n + \alpha_n(y_{ij}^n-x_{ij}^n)$
6. Convergence Test
   If the convergence criterion is met stop; otherwise go to step 2.

Formulation of the Assignment Problem with Pointwise Flow-Delay Relationships

As mentioned earlier, the objective of this work is to develop an assignment methodology not based on aggregate and simplified flow delay relationships. Let *FDM* be the delay vector produced by a flow delay model with unknown mathematical characteristics or by a simulation model:

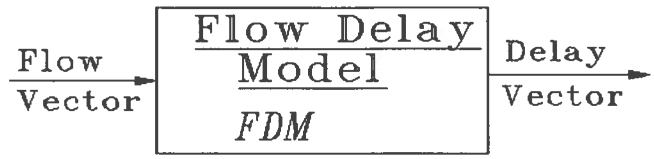

Figure 2: Example of a Pointwise Flow Delay Model

When dealing with such a function, it is impossible to evaluate the objective function of the following equilibrium assignment problem:

$$\min \quad f(x) = \sum_{ij} \int_0^x FDM(w)\, dw \tag{14}$$

One possibility to overcome this problem is to estimate a new flow delay relationship based on the results of the simulation values. This approach was adopted by the developers of several solution algorithms SATURN [12] being one of them.
When applying Leblanc's [7] algorithm directly to solve the problem of Equation (7) there are two steps of the algorithm which may be problematic to solve, (a) the solution of the linear program, Equation (10) and (b) the one-dimensional search, Equation (11). Assuming that the *FDM* function represents an underlying continuous and nonotonic non decreasing function the LP part of the original Leblanc's algorithm can be easily applied. It can easily be shown that no problem arises by the use of *FDM* in the LP problem since the term:

$$\min \quad \sum_{ijk} \frac{\partial f(\boldsymbol{x}^1)}{\partial x_{ij}^s} y_{ij}^s \tag{15}$$

reduces to the following one:

$$\min \sum_{ijk} FDM(x_{ij})\, y_{ij}^{s} \qquad (16)$$

The line search step for the optimal move size (Equation (11)) can not be solved easily using *FDM* model. The Line Search step of Leblanc's algorithm requires a continuous evaluation of the objective function (equation (14)) in order to find its minimum. This can not be done since the functions are unknown analytically and thus the function's integral is not known.

Solution Algorithms

As shown in the previous section the line search step can not be implemented directly. At each iteration of the assignment algorithm, the new solution x^{n+1}, lies between x^{n} (the old solution) and y^{n}. The new point can be calculated as:

$$\boldsymbol{x}^{n+1}=\boldsymbol{x}^{n}+\alpha\,(\boldsymbol{y}^{n}-\boldsymbol{x}^{n}) \qquad (17)$$

Which is equivalent to

$$\boldsymbol{x}^{n+1}=(1-\alpha)\,\boldsymbol{x}^{n}+\alpha\boldsymbol{y}^{n} \qquad (18)$$

of this research the optimum value of α (optimal move size) can not be determined using the method proposed by Leblanc, thus another linear combination method and has to be applied. Before the proposed method is presented, a solution method of successive averages - MSA, suggested first by Sheffi [10] is discussed.

Successive Averages Method

The method of successive averages (MSA) is based on stochastic approximation methods. Stochastic approximation is concerned with the convergence of problems which are stochastic in nature usually based on observations which involve errors. Search techniques which successfully reach an optimum in spite of the noise have been 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 "goal", while the term stochastic suggest the random character of the function being evaluated.

The Robbins Monroe procedure places solution point *n+1* according to the solution of point *n*

$$x_{n+1}=x_n+\alpha z(x_n) \tag{19}$$

where *z(x)* is a "noisy" function. The method is based on predetermined move sizes, α, that has to satisfy the following two conditions:

$$\begin{aligned}&\sum_{n=1}^{\infty}\alpha_n\rightarrow\infty\\&\sum_{n=1}^{\infty}\alpha_n^2<\infty\end{aligned} \tag{20}$$

One of the simplest step-size sequences, that satisfy both conditions is the sequence:

$$\alpha_n=\frac{1}{n} \tag{21}$$

In general, any sequence such that:

$$\alpha_n=\frac{K_1}{K_2+n} \tag{22}$$

where K^1 is a positive constant and K^2 is a nonnegative constant can be used.

Sheffi [10] applied this methodology to solve a probabilistic assignment problem. This approach can also be applied to the solution of deterministic equilibrium assignment. The whole algorithm can be summarized as:

1 Initialization

(1) Run the simulation program with an initial flow vector and (2) perform an *All or Nothing assignment*. This yields to flow vector x^1

2 Update Travel Times

Perform a simulation run with flow vector x^n, this yields t_{ij}^n

3 Direction Finding

Perform an *All or Nothing assignment* with t_{ij}^n. This yields y_{ij}^n

4 Next Point

Find a point x^{n+1} between x^n and y^n.

$$x^{n+1}=x^{n}+\frac{1}{n}(y^{n}-x^{n}) \qquad \textbf{(23)}$$

Increase iteration counter n.

5 Convergence Test

If the convergence criterion is met stop; otherwise go to step 2.

The drawbacks of the algorithm with predetermined step sizes is that its convergence is very slow, and it is difficult to design appropriate convergence criteria [9].

The slow convergence of this methodology is not the only problem of the Moving Averages Method. The MSA algorithm was applied to solve the assignment problem of a network consisting of three links and one OD pair (see Sheffi [10] page 114). Figure 3 shows the

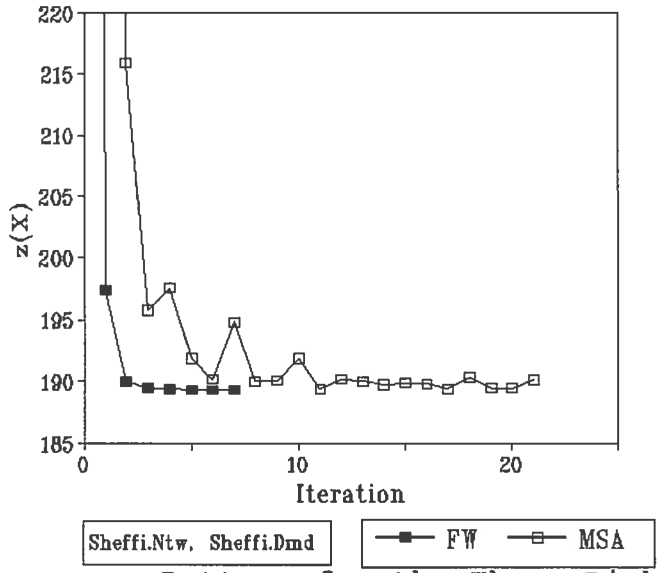

Figure 3: Convergence Pattern for the Three Link Network

objective function, *z(x)*, as function of the iteration number. It can be seen, that if the assignment procedure is ended after a predetermined number of iterations, a solution with bad convergence characteristics may be chosen. This occurs due to the fact that the convergence of the MSA method is not asymptotic, but it oscillates around the approximate solution. Furthermore the MSA method is suppose theoretically to converge under certain regularity conditions (Powell & Sheffi [9]). However numerical computer roundoff errors might be quite significant when the number of iterations is high. This errors and the small difference in links loads from one iteration to the other when n is high might prevent this algorithm to converge to the correct solution.

Linearization Method

Due to the drawbacks of the MSA method. A new methodology by means of which any *FDM* function can be used to solve the equilibrium traffic assignment was searched. As mentioned in previous sections, a number of methodologies exist which take the delay values from simulation models. A simulation model can be considered a *FDM* function. We can obtain the delay of traffic on any link or turn movement on the network for a given traffic flow pattern. But it is impossible to do further mathematical manipulations on the relation between flow and delay.

The proposed method is based on a linear approximation of the real flow-delay function. At each iteration of Frank-Wolfe's algorithm we generate a new flow-delay pair for each network element and calculate a straight line which paths through the previous flow-delay pair and the present one. For errorless *FDM* function this straight line will always be a non decreasing function with volume. the succession of this straight lines and Frank-Wolf iterations are the basic iterations of the proposed algorithm. Theoretically it is possible to fit a curve based on all the flow-delay pairs obtained during the assignment process. This is, however, a cumbersome work which requires large storage space and its advantage is not clear when the actual shape of the *FDM* function is not known. Therefore we chose the simplest of all approximations, the linear one. At each iteration of Frank-Wolfe's algorithm, only two (<flow>,<delay>) pairs are considered. At iteration n of the algorithm the straight line defining the present flow-delay relationships is based on the x^{n-1} and x^{n} values. the practical implication of this approach is that at any point in the algorithm only one set of <flow>,<delay> points needs to be stored. An example of linear relationship at each iteration are presented in Figure Figure 4.

Mathematically the linear flow-delay relations can be expressed as follows:

$$t=\theta_{0i}+\beta_{i}x_{i} \qquad \textbf{(24)}$$

Obviously, if this is the relationship between the flow and the delay there are no problem in the implementation of Frank-Wolfe's algorithm.

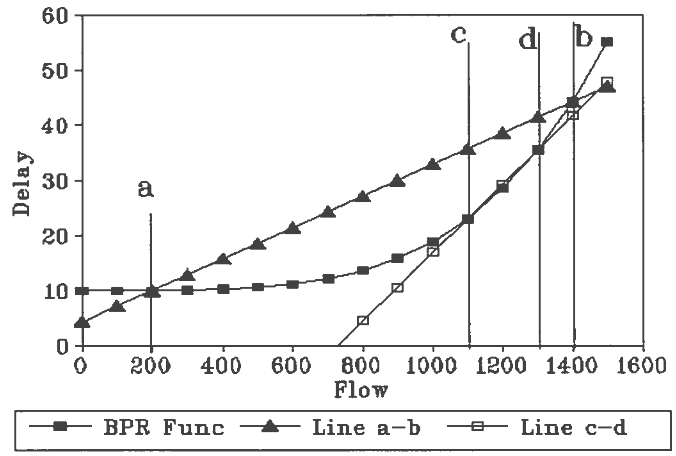

Figure 4: Linearized Flow Delay Relationship

The temporal (for the current iteration) objective function is:

$$\min \quad z(x) = \sum_{ij} \int_0^x y(w)\, dw \qquad \textbf{(25)}$$

And can be expressed as:

$$\min \quad z(x) \sum_{ij} \int_0^x \left[\theta_{ij} x_{ij} + \frac{b_{ij}}{2} x_{ij}^2 \right] \qquad \textbf{(26)}$$

The step that could not be solved when using pointwise flow-delay functions (*FDM*), can now be easily implemented. Moreover, when using a linear functions the optimal move size can be calculated in an exact manner and no line search method is required. This improves computer running time of each iteration of the algorithm. Given two feasible flow vectors, $\underline{\mathbf{x}}$ and $\underline{\mathbf{y}}$, the line search step determines the minimum of the original function along the line between the two flow vectors. In the case of a linearized function, the objective function is convex with respect to x_{ij}, meaning that there exist unique minimum in the interval between $\underline{\mathbf{x}}$ and $\underline{\mathbf{y}}$.
The step size can be calculated according to the following expression:

$$\min \; z\,[\boldsymbol{x}^n+\alpha(\boldsymbol{y}^n-\boldsymbol{x}^n)] \qquad \textbf{(27)}$$
$$st: \quad 0\leq\alpha\leq 1$$

Defining $\underline{\mathbf{d}}^n$ as the direction between $\underline{\mathbf{x}}^n$ and $\underline{\mathbf{y}}^n$ ($\underline{\mathbf{d}}^n = \underline{\mathbf{y}}^n - \underline{\mathbf{x}}^n$) equation(27) can be expressed as:

$$\min \; z(\boldsymbol{x}+\alpha\boldsymbol{d}) = \min \; \sum_i \left[\theta_i(x_i+\alpha d_i)+\frac{\beta_i}{2}(x_i+\alpha_i d_i)^2\right] \qquad \textbf{(28)}$$

The optimal step size, α, can be analytically determined according to the following expression:

$$\alpha = -\frac{\sum_{ij}(\theta_{ij}d_{ij}+\beta_{ij}x_{ij}d_{ij})}{\sum_{ij}\beta_{ij}d_{ij}^2} \qquad \textbf{(29)}$$

Using the linearized function, $z(.)$, and the step size, α, Frank-Wolfe's 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 can be achieved.

The proposed algorithm can be summarized as follows:

1. Initialization
   (1) Calculate an initial delay vector based on *FDM*.
   (2) Perform an *All or Nothing assignment*. This yields to flow vector $\underline{\mathbf{x}}^1$.
2. Update Travel Times
   Calculate the delay vector with flow vector $\underline{\mathbf{x}}^n$.
   $FDM(\underline{\mathbf{x}}^n) = \underline{\mathbf{t}}^n$.
3. Linearization
   Calculate the linearized function z(x) based on vectors $\underline{\mathbf{x}}^{n-1}$ and $\underline{\mathbf{x}}^n$.

4. Direction Finding

   Perform an *All or Nothing assignment* with $\underline{\mathbf{t}}^n$. This yields the vector $\underline{\mathbf{y}}^n$.

5. Next Point

   (1) Calculate the step size according to Equation (29).

   (2) Set $\underline{\mathbf{x}}^{n+1} = \underline{\mathbf{x}}^n + \alpha\ (\underline{\mathbf{y}}^n - \underline{\mathbf{x}}^n)$.

   (3) Increase iteration counter n.

6. Convergence Test

   If the convergence criterion is met stop; otherwise go to step 2.

Examples and Results

To determine the ability of the proposed algorithm to provide accurate estimates of the traffic flow vector, the method was tested with three different networks. For each network different flow-delay relationships were assumed. These flow-delay relationships were based on the BPR functions [1], equation (1) with different α and ß values.

The proposed assignment methodology was compared to two existing assignment methodologies: Leblanc's implementation of Frank-Wolfe's decomposition algorithm and Sheffi's method of successive averages (MSA). The proposed methodology was 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 function was evaluated at discrete points, as if it is not possible to calculate the original objective function integral - $\int t(w)dw$.

The proposed method was applied initially to the three links network given by Sheffi. Figure 5 shows the convergence pattern for the three methods, when applied to the three link network given by Sheffi [10]. It can be seen that the proposed method converges asymptotic 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 and second, the number of iterations necessary to achieve acceptable solution is significantly smaller.

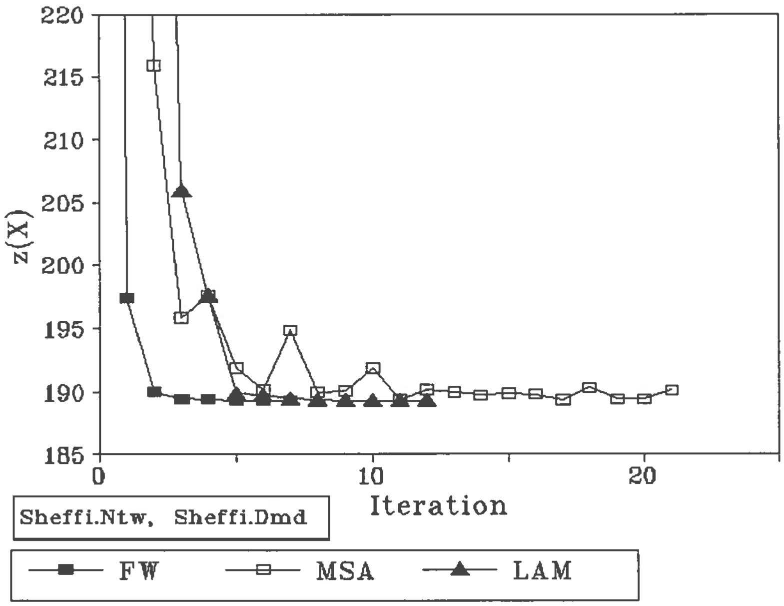

Figure 5: Convergence Pattern for the Three Link Network

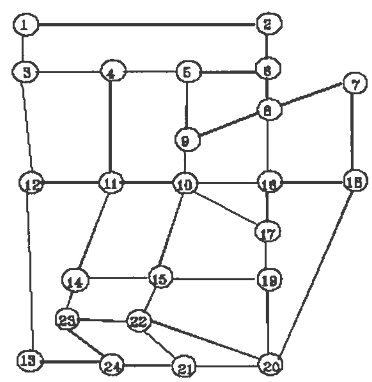

Figure 6: Sioux Falls Network (Leblanc [7])

The method was also applied to a nine link and a 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 the "classic" Sioux Falls network, presented in the original work by Leblanc [10] (see figure 6). This is a 24 nodes, 76 links network. Several assignment runs with different BPR volume-delay curves with where performed. The different α and β values of the delay curves where changed to examine the behavior of the assignment algorithms under various congestion conditions. Twenty five iterations of the proposed algorithm and the MSA method were performed for each

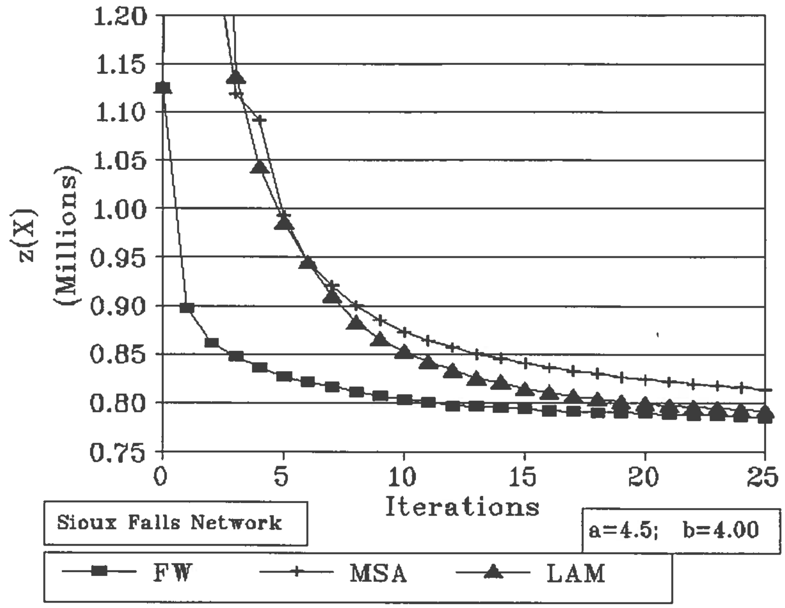

Figure 7: Convergence Pattern for the Sioux Falls Network

volume delay curve. As expected the proposed method gave better results than the MSA method. After 25 iterations of the algorithm, the proposed method was always closer to the exact solution obtained by means of Leblanc's algorithm. Table I shows the results for various combinations of the BPR model parameters. It can be seen that, no matter what kind of flow-delay model is used, the proposed model's results were closer to the exact solution than those of the MSA method. Further more the convergence characteristics of the proposed method don't deteriorate when sensitivity, of the network elements, to congestion increase. Observe in table I that this doesn't seem to be the case for the MSA assignment procedure.

Conclusions

The proposed linearization assignment methods seems to work very well. When a errorless deterministic *FDM* exists the proposed method is clearly superior to the MSA method. One of the big advantages of the proposed method is that it provides an elegant simple and computer storage efficient iterative procedure to perform traffic assignment when the volume-delay curves are not explicitly specified. It can easily be adopted to situation where part of the network elements are represented by volume-delay curves while the behavior of others is determined by *FDM* functions. Furthermore this method seems well suited to be applied as a second refined assignment stage using as a staring points the solution vector generated based on aggregate crude volume-delay functions. Procedures which perform stochastic assignment are of great interest lately. The ability of the proposed procedure to perform stochastic assignment was not fully investigated. One of the problems which might arise when applying the proposed method to stochastic assignment is that the slope of the straight line generated at some iteration of the algorithm might be negative. This will indicate a decrease of travel time with volume and might although not necessarily will imperil the convergence of the procedure. A way to over come this problem can be simply assigning a zero slope or using the previously calculated slope when such a problem occurs. The convergence characteristics of the proposed method when performing stochastic assignment need further investigation.

References

[1.] *Traffic Assignment Manual*, Bureau of Public Roads, Urban Planning Division, US Department of Commerce, Washington DC., 1964

[2.] R. AKCELIK. The Highway Capacity Manual Delay Formula for Signalized Intersections. *ITE Journal*, 58:23--27, 1988.

[3.] M. AVRIEL. *Nonlinear Programming: Analysis and Methods*. Prentice-Hall, Inc., 1976.

[4.] M. BECKMAN, C.B McGUIRE, and C.B WINSTEN. *Studies in the Economics of Transportation*. Yale University Press, 1956.

[5.] K.B. DAVIDSON. A Flow Travel Time Relationship for Use in Transportation Planning. *Proceedings, Australian Road Research Board*, 3:183--194, 1966.

[6.] M.D. HALL, D. VAN VLIET, and L.G. WILLUMSEN. SATURN - A Simulation-Assignment Model for the Evaluation of Traffic Management Schemes. *Traffic Engineering & Control*, pages 168--175, 1980.

[7.] L.J. LEBLANC. *Mathematical Programming Algorithms for Large Scale Network Equilibrium and Network Design Problems*. PhD thesis, Northwestern University, 1973.

[8.] J. ORTUZAR and L.G. WILLUMSEN. *Modelling Transport*. John Wiley & Sons, 1990.

[9.] W.B. POWELL and Y. SHEFFI. The Convergence of Equilibrium Algorithms with Predetermined Step Sizes. *Transportation Science*, 16(1):45--55, 1982.

[10.] Y. SHEFFI. *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*. Prentice-Hall, Inc., 1985.

[11.] Y.J. STEPHANEDES, I.Z. ARGIROPOULOS, and P. MICHALOPOULOS. On-Line Traffic Assignment for Optimal Freeway Corridor Control. *Journal of Transportation Engineering*, 116(6):744--755, 1990.

[12.] D. VAN VLIET. SATURN - A Modern Assignment Model. *Traffic Engineering & Control*, pages 578--581, 1982.

[13.] F.V. WEBSTER. *Traffic Signal Settings*. Her Majesty's Stationary Office, 1958. Road Research Laboratory Technical Paper 39.