HEURISTIC DYNAMIC ASSIGNMENT BASED ON MICROSCOPIC TRAFFIC SIMULATION

Jaime Barceló¹ and Jordi Casas²

¹Dept. of Statistics and Operations Research, Universitat Politecnica de Catalunya, Barcelona, Spain, jaume.barcelo@upc.es,
²TSS-Transport Simulation Systems, Barcelona, Spain, casas@tss-bcn.com, http://tss-bcn.com

ABSTRACT

A common drawback to most of the DTA models is that they do not represent properly spillback of congestion. Simulation has been suggested as an approach to overcome the problem. The solution approach to DTA would then consist of two components, an analytical method to determine the path dependent flow rates and a simulation based network loading to determine arc volumes. The improvements in software and hardware technologies have made possible to simulate microscopically real networks of sensitive size. This paper describes a heuristic approach to DTA in which two alternative analytical components are used to determine the path flow rates, one based on a stochastic route choice method, and another one based on an approximation to dynamic user equilibrium conditions, while the network loading is done by a microscopic simulation model.

1 INTRODUCTION

According to Florian et al., (2001), a dynamic traffic assignment model consists of two main components:

1. A method to determining the path dependent flow rates on the paths on the network, and
2. A Dynamic Network Loading method, which determines how these path flows give rise to time-dependent arc volumes, arc travel times and path travel times

The diagram in Figure 1 depicts the logic conceptual approach for dynamic traffic assignment models.

Path flow rates depend on the emulation of path choice behavior of drivers. Two alternative approaches are going to be considered in this paper:

- Dynamic assignment en route: at each time period the corresponding fraction of the demand is assigned to the currently available paths for each Origin-Destination pair according to the probabilities estimated by a route choice model. Driver can be allowed to dynamically change the route en route if a better path from their current position to their destination is available.
- Dynamic Equilibrium Assignment: path flows are determined by an approximate solution to the mathematical model for the dynamic equilibrium conditions.
The Dynamic Network Loading, also known as Dynamic network Flow Propagation, Cascetta (2001), “models simulate how the time-varying continuous path flows propagate through the network inducing time-varying in-flows, out-flows and link occupancies”. A wide variety of approaches, from analytical, Wu et al. (1998a), Xu et al. (1998), Xu et al. (1999), to simulation based, Florian et al. (2001).

A Dynamic Network Loading mechanism based on the AIMSUN microscopic simulation model, Barceló et al. (1995), (1998), (1999), is proposed in this paper. This traffic simulation approach is proposed not only due to its ability to capture the full dynamics of time dependent traffic phenomena, but also for being capable of dealing with behavioral models accounting for drivers’ reactions in the way required by the Dynamic Network Loading. However, it should be noticed that the simulation of such systems requires a substantial change in the traditional paradigms of microscopic simulation, in which vehicles are generated at the input sections in the model, and perform turnings at intersections according to probability distributions. In such models vehicles have neither origins nor destinations and move randomly on the network. The required simulation approach should be based on a new macroscopic simulation paradigm: a route based microscopic simulation.

In this approach, vehicles are input into the network according to the demand data defined as an O/D matrix (preferably time dependent) and they drive along the network following specific paths in order to reach their destination. In the main Route Based simulation new routes are to be calculated periodically during the simulation, and a Route Choice model is needed, when alternative routes are available.

![Figure 1. Conceptual approach to dynamic traffic assignment](image-url)
2 ESTIMATION OF PATH FLOW RATES: ASSIGNMENT BASED ON DISCRETE ROUTE CHOICE MODELS

The vehicles follow paths from their origins in the network to their destinations. So the first step in the simulation process is to assign a path to each vehicle when it enters the network. In the implementation done in AIMSUN the candidate paths can be of different types:

- **User-defined Paths (UdP):** Predefined using the network editor or as an output from other traffic simulators or transportation models, either macroscopic (i.e. a transport planning) or microscopic.

- **Calculated Shortest Path Trees (CSPT):** Shortest path trees calculated using the default or user defined cost functions. There are two types of CSPT:
  - **Initial Shortest Path Tree (ISPT):** For each destination centroid it provides a shortest path tree, using an initial default or user defined cost function for each turning movement.
  - **Statistical Shortest Path Tree (SSPT):** For each destination centroid and every user defined time interval it provides a shortest path tree, using the values of the cost function for each turning movement for that time interval. The value of the cost function is updated for each time interval on basis to the statistical data previously gathered during the simulation.

A vehicle of vehicle type \( vt \) going from origin \( O_i \) to destination \( D_j \), can choose one path from the following discrete choice set of alternative paths:

- \( N \) User-defined Paths: \( UdP_k(O_i, D_j) \quad k=1..N \)
- \( I \) Initial Shortest Path Tree: \( ISPT(D_j) \)
- \( P \) Statistical Shortest Path Trees: \( SSPT_k(D_j) \quad k=1..P \)

With probabilities

\[
P(UdP_k(O_i, D_j), vt) : \text{Probability of use } UdP_k(O_i, D_j) \text{ by a vehicle type } vt
\]

\[
P(ISPT(D_j), vt) : \text{Probability of use } ISPT(D_j) \text{ by a vehicle type } vt
\]

Satisfying the condition:

\[
\sum_{k=1}^{N} P(UdP_k(O_i, D_j), vt) + P(ISPT(D_j), vt) \leq 1
\]

At the beginning of the simulation, shortest path trees are calculated from every section to each destination centroid, taking as arc costs the specified initial costs. During the simulation, new routes are recalculated in every time interval, taking the specified arc costs updated for each arc after the statistics gathered during a number of statistics gathering interval defined by the analyst.

The user may define the time interval for recalculation of paths, that is the frequency at which paths are refreshed according to the prevailing traffic conditions, and the maximum number of path trees to be maintained during the simulation. When the maximum number of path trees (K) is reached, the oldest paths will be removed as soon as no vehicle is following them. It is assumed that vehicles only choose between the most recent K path trees. Therefore, the oldest ones will become obsolete and disused.
From the point of view of the modeling approach there are two alternatives determining how the dynamic network loading will be performed:

- The concept of cost used in updating the routes
- The route choice model used in assigning vehicles to available routes

Assuming that route cost is the sum of the costs of the arcs composing the route, a wide variety of arc costs can be proposed: travel times at each simulation interval, toll pricing, historical travel times representing driver’s experience from previous days, combinations of various arc attributes as for instance travel times, length and capacity, etc.

The version 4.1 of AIMSUN, GETRAM/AIMSUN (2001), provides the user with two alternatives: use default arc costs or use the Function Editor included in the TEDI set of graphical editors of the GETRAM traffic modelling environment to define his/her own arc cost function using as arguments any of the numerical attributes of the road sections, statistical values or vehicle characteristics. Calculation of shortest paths is carried out per vehicle type, taking into account reserved lanes. Therefore, the set of paths from which a vehicle may select, either when entering the network or when being re-routed, may be different for different vehicle types even though they travel to the same destination, depending on the presence of reserved lanes. Also the travel time used in the cost function for recalculation of shortest paths is taken as the travel time per vehicle type. The default cost assigned to each arc is a function of the travel time of the section and the turning movement, the arc capacity can also be taken into account and then the default cost of arc $a$ for vehicle type $vt$ is calculated as:

$$Cost(a,vt) = CurrentCost(a,vt) + CurrentCost(a,vt) \times \phi \times \left(1 - \frac{\text{Capacity}(a)}{\text{MaxCapacity}}\right)$$

As an alternative to the Default Initial and Cost Functions users can define their own Cost Functions. This is done via the Function Editor in Tedi. If no User-Defined Function is assigned to an arc, the Default Cost Function is applied. To define an Initial or Cost Function, the user can use any of the most common mathematical functions and operators (+, -, *, /, ln, log, exp, etc.). The function parameters can be constants or variables that are data related to the description of the network, sections, turning movements and vehicle types. For Cost Functions, as there is also simulated output data available, variables corresponding to simulated statistical data can also be used in the cost function.

Figure 2, illustrates an example of a simulation model for which alternative arc cost functions have been defined by the user to define alternative simulation scenarios and test which fits more suitably the actual traffic conditions. The open window shows part of the algebraic expression for a cost function per vehicle type.

In a similar way when using these dynamic assignment abilities it could be raised the question of which is the most suitable route choice function. Route choice functions represent implicitly a model of user behaviour, representing the most likely criteria employed by the user to decide between alternative routes: perceived travel times, route length, expected traffic conditions along the route, etc. The solution implemented in the version 4.1 of AIMSUN also provides the user two alternatives: use the default functions or define his/her own route choice function by means of the Function Editor.
The most used route choice functions in transportation analysis are those based on the discrete choice theory, i.e. Logit functions assigning a probability to each alternative route between each origin-destination pair depending on the difference of the perceived utilities. A drawback reported in using the Logit function is the exhibited tendency towards route oscillations in the routes used, with the corresponding instability creating a kind of flip-flop process. According to our experience there are two main reasons for this behavior. The properties of the Logit function and the inability of the Logit function to distinguish between two alternative routes when there is a high degree of overlapping.

![Function Editor Menu](image-url)

Figure 2. Example of User defined arc cost function per vehicle type

The instability of the routes used can be substantially improved when the network topology allows for alternative routes with little or no overlapping at all, playing with the shape factor of the Logit function and re-computing the routes very frequently. However, in large networks where many alternative routes between origin and destinations exist and some of them exhibit a certain degree of overlapping (see Figure 3), the use of the Logit function may still exhibit some weaknesses.

![Diagram of Overlapping Routes](image-url)

Figure 3: Overlapping Routes
To avoid this drawback the C-Logit model, Cascetta et al. (1996), Ben-Akiva et al. (1999), has been implemented. In this model, the choice probability $P_k$ of each alternative path $k$ belonging to the set $I_{rs}$ of available paths connecting an O/D pair, is expressed as:

$$P_k = \frac{e^{\theta(V_k - CF_k)}}{\sum_{i \in I_{rs}} e^{\theta(V_i - CF_i)}}$$

where $V_i$ is the perceived utility for alternative path $i$, and $\theta$ is the scale factor, as in the case of the Logit model. The term $CF_k$, denoted as ‘commonality factor’ of path $k$, is directly proportional to the degree of overlapping of path $k$ with other alternative paths. Thus, highly overlapped paths have a larger CF factor and therefore smaller utility with respect to similar paths. $CF_k$ is calculated as follows:

$$CF_k = \beta \cdot \ln \left( \frac{\sum_{i \in I_{rs}} \left( \frac{L_{ik}}{L_{i}^{1/2} L_{k}^{1/2}} \right)^{\gamma}}{\gamma} \right)$$

where $L_{ik}$ is the length of arcs common to paths $i$ and $k$, while $L_i$ and $L_k$ are the length of paths $i$ and $k$ respectively. Depending on the two factor parameters $\beta$ and $\gamma$, a greater or lesser weighting is given to the ‘commonality factor’. Larger values of $\beta$ means that the overlapping factor has greater importance with respect to the utility $V_i$; $\gamma$ is a positive parameter, whose influence is smaller than $\beta$ and which has the opposite effect.

To get the insight on what is happening in a heuristic dynamic assignment for the proper calibration and validation of the simulation model the user should have access to the analysis of the used routes. To support the user in this analysis process AIMSUN includes a path analysis tool. Figure 4 depicts the path dialogue window.

The path list box contains the list of section identifiers composing the path and the following information is displayed for each section:

- The cost in time (seconds) from each of the sections in the path to the destination centroid. This can be calculated as either the sum of $IniCost(a)$ or $Cost(a, vt)$ of all the arcs composing the path.
- The travel time in seconds from each of the sections in the path to the destination centroid. This is equal to the cost only if the capacity weight parameter is set to zero.
- The distance (metres) from each of the sections in the path to the destination centroid.

In the example shown in Figure 4, the shortest path from section 1 to centroid 11 goes through sections 14, 15, 10, 11 and 12. The cost of the whole path is 247.9 seconds, the travel time is 139.5 seconds and the distance is 655.4 meters. In this case Cost and Travel Time are different, as the Capacity Weight has been set to 1.25.
Vehicles are initially assigned to a route from a set of available routes on a probabilistic way determined by the selected route choice model. The following algorithm defines initial assignment path of a vehicle $v$, which belongs to vehicle type $vt$, and goes from origin $Oi$ to destination $Dj$:

$\text{guided} = \text{GuidedBehaviour}(v)$

\begin{align*}
&\text{if} \text{guided} \text{ then} \\
&(CO_i, CD_j) = \text{DecideConnectionsTo} & \text{& From}(O_i, D_j) \\
&\text{Assigned Path} = \text{RouteChoice}((CO_i, CD_j), ISPT(D_j), SSPT_1(D_j), \ldots, SSPT_p(D_j))
\end{align*}

\begin{align*}
&\text{else} \\
&\text{index} = \text{GiveIndex}(P(ISPT(D_j), vt), P(UdP_1(O_i, D_j), vt), \ldots, P(UdP_N(O_i, D_j), vt)) \\
&\text{if} \text{index} = -1 \text{ then} \\
&(CO_i, CD_j) = \text{DecideConnectionsTo} & \text{& From}(O_i, D_j) \\
&\text{Assigned Path} = \text{RouteChoice}((CO_i, CD_j), ISPT(D_j), SSPT_1(D_j), \ldots, SSPT_p(D_j))
\end{align*}

\begin{align*}
&\text{else} \\
&\text{if} \text{index} \leq N \text{ then} \\
&\text{Assigned Path} = UdP_{\text{index}}(O_i, D_j) \\
&\text{else} \\
&\text{Assigned Path} = \text{RouteChoice}((CO_i, CD_j), ISPT(D_j)) \\
&\text{endif}
\end{align*}

\text{endif}

where
- **GuidedBehaviour** returns true in the case of vehicle $v$ is guided and generates a random number between 0 and 1, if it is less than the guidance acceptance of $v$. It return false otherwise,

- **DecideConnectionsTo&From** returns the set of connectors available to use a vehicle to enter in the system and go out. $CO_i$ is the set of connectors available from origin centroid $Oi$ and $CD_j$ is the set of connectors available from destination centroid $D_j$. In the case of having a centroid with “consider percentages” enabled, the connector to enter or exit is chosen randomly, and in the case of having a centroid with “consider percentages” disabled, all connectors are considered.

- **RouteChoice** function, returns the path assigned taking into account the distribution of the route choice model chosen by the user.

- **GiveIndex** returns the index of assigned probability, given a set of probabilities $Pi$, where the total sum is less or equal than 1. If this index is –1, means no path is chosen.

Apart from the initial assignment of route, which is made at the vehicle’s departing time, there is the possibility of making a route reassignment during the trip. This is the Dynamic route choice model in which a vehicle can make a new decision about what route to follow at any time along its trip, whenever there are new shortest routes available. Whereas in the Static model, a vehicle will always follow its initially selected route until reaching the destination, although new shortest route could be available during the trip.

In the Dynamic model all vehicles or only a special class, that of the guided vehicles can take the decision of changing to a new shortest route during the trip, assuming that this information is available to them. The simulation process based on the time dependent routes consists of the following steps which logic is illustrated in the diagram in figure 5.

1. **Calculate initial shortest routes for each O/D pair using the defined initial costs.**
2. **Simulate for a predefined period (e.g. 5 minutes) assigning to the available routes the fraction of the trips between each O/D pair for that period according to the selected route choice model and obtain new average link travel times as a result of the simulation.**
3. **Recalculate shortest routes, taking into account the current average link travel times.**
4. **If there are guided vehicles, or variable message signs suggesting rerouting, provide the information calculated in 3 to the drivers that are dynamically allowed to reroute.**
5. **Go to step 2.**
3 ESTIMATION OF PATH FLOW RATES: ROUTE CHOICE BASED ON DYNAMIC EQUILIBRIUM ASSIGNMENT

The preventive formulation of the dynamic user equilibrium problem considers that paths are chosen by the drivers, this leads to models describing the evolution of the path flows when users make route choice decisions based on the actual experienced travel times. The temporal version of Wardrop’s principle can then be stated in the following terms: “The feasible path flow rates \( h \) on \( \Omega \) with well defined actual travel times \( s \) are said to be in a user optimal
dynamic equilibrium if, at instant \( t \), the actual path travel time is minimised, up to a set of measure zero, when the path flows are positive”. The problems can then be stated in mathematical terms in the space of the path flows \( h_k(t) \) in terms of the corresponding user optimal equilibrium conditions which are:

\[
\begin{align*}
S_k(h^*(t)) = u_i(t), \text{ if } h_k^*(t) > 0 \\
\geq u_i(t) \text{ otherwise }
\end{align*}
\]

where \( h_k^*(t) \in \Omega \) and \( u_i(t) = \min_{k \in K_i} \{S_k(h^*(t))\} \text{ for almost all } t \in [0,T] \). 

The path flow rates satisfy conservation of flow and are nonnegative, conditions that define the set \( \Omega \) of feasible flows as:

\[
\Omega = \left\{ h(t): \sum_{k \in K_i} h_k(t) = g_i(t), \forall i \in I; h_k(t) \geq 0, \forall k \in K_i \right\} 
\]

for almost all \( t \in [0,T] \). 

where \( h_k(t) \) is the flow at time \( t \) on path \( k \) belonging to the set \( K_i \) of all paths for the \( i \)-th origin-destination pair, for the set \( I \) of all O-D pairs, and \( g(t) \) is the demand for time \( t \). 

Friesz et al. (1993) have shown that the statement of the user optimal dynamic equilibrium problem in the space of the path flows, if \( g(\cdot) \) is Lebesgue integrable on \([0,T]\), and \( S(\cdot) \) is continuously differentiable in \( v \) is equivalent to the variational inequality problem of finding \( h^* \in \Omega \) such that:

\[
\int \sum_{I \in I} \sum_{k \in K_i} S_k(h^*(t))h_k(t) - h_k^* dt \geq 0
\]

This problem can be approximated by a discretization scheme, Wu et al. (1998b), to solve the related finite dimensional variational inequality problem

\[
(S(h^*), h - h^*) \geq 0, \forall h \in \Omega
\]

The discretization scheme consists on dividing the time horizon \([0,T]\) into a finite number of time periods of length \( \Delta t \), \( M = \{1,2,\ldots,m\} \), \( m = \frac{T}{\Delta t} \). The problem is then solved by an adaptation of a projection method for each time period that uses the Fukushima gap, Fukushima (1992) and solves at iteration \( l \) the auxiliary quadratic optimization problem:

\[
\min_{h \in \Omega} \left[ (h - h^T) S(h^T) + \frac{1}{2\alpha} (h - h^T)^T S(h - h^T) \right]
\]

which can be decomposed into a collection of smaller, independent, convex quadratic programs, one for each origin-destination pair, the program for the \( i \)-th O-D pair is:
Two algorithmic approaches for solving problem (6):

A. Direct projection method

The problem at iteration \( l \) finds the flow rates \( h^{l+1} \) to be used at iteration \( l+1 \) in the Dynamic Network Loading process. To formulate the problem first one has to find at the end of iteration \( l \) if there is a new shortest path \( k \) with cost \( \tilde{s}_k < \min_{k \in K^l_i} s_k(h^l) \) (\( \tilde{s}_k \) computed with the link costs \( s_a(v^l) \)), then the path \( k \) is added to the set of paths \( K^l_i \) used at iteration \( l \): \( K^{l+1}_i = K^l_i \cup \{k\} \).

The linear constraint in (6) can be written as \( a^{l+1}_i h = g^{l+1}_i \) where \( a^{l+1}_i = (1,1,\ldots,1) \in \mathbb{R}^{K^l_i} \) a feasible descent direction for the objective function in (6) can be found projecting the gradient of the objective function onto

\[
\mathfrak{K} = \left\{ h \in \mathbb{R}^{K^l_i} \middle| a^{l+1}_i h = g^{l+1}_i; h \geq 0 \right\}.
\]

The projection vector is \( y' = -P \nabla f(h') \), when \( P = I - a(a^T a)^{-1}a \) is the projection matrix, thus:

\[
f(h) = \sum_{k \in K^{l+1}_i} \left[ (h_k - h'_k) s_k(h^l) + \frac{1}{2\alpha} (h_k - h'_k)^2 \right] \Rightarrow \nabla_k f(h) = \frac{\partial f(h)}{\partial h_k} = \frac{1}{\alpha} h_k + s_k(h^l) - \frac{1}{2\alpha} h'_k^2
\]

\[
\nabla_k f(h') = \frac{1}{\alpha} h'_k + s_k(h^l) - \frac{1}{2\alpha} h'_k^2 = \frac{1}{2\alpha} h'_k + s_k(h^l), \quad k \in K^{l+1}_i
\]

and therefore.

\[
h^{l+1} = h' + \lambda' \left[ h' - y' \right]^r
\]

where the optimal value of the step length \( \lambda' \) is:

\[
\lambda' = \frac{g^{l+1}_i - g^{l}_i}{g^{l}_i - \sum_{k \in K^{l+1}_i} y'_k}
\]
B. Method of Successive Averages

1. Start with one shortest path and add another (new shortest path) on each iteration (n), up to some pre-specified maximum number (N).
2. On each iteration reduce the flow on the existing paths by a fraction of \(1/n\), and give this flow to the new path.
3. After reaching iteration \(N\), stop adding new paths, but continue to use the iteration number \((1/n)\) to calculate how much flow will be moved from the non-shortest paths to the shortest path.

**Iteration \(l\):**

- Let \(g_{i}^{(l+1)}\) the amount of flow for the \(i\)-th O-D pair to enter the network at iteration \(l\).
- Let \(K_{i}^{(l+1)} = K_{i}^{(l)} \cup \{\tilde{k}\}\) the set of paths for the \(i\)-th O-D pair at iteration \(l\), where \(\tilde{k}\) is a new shortest path with cost \(\tilde{s}_{\tilde{k}} < \min_{\tilde{k} \in K_{i}^{(l)}} S_{\tilde{k}}(h_{\tilde{k}})\) (cost \(\tilde{s}_{\tilde{k}}\) computed with the link costs \(s_{a}(v)\)).
- When \(|K_{i}^{(l)}| = N\) then \(\tilde{k}\) is the shortest of the \(N\) paths with the new costs.
- The new flows \(h_{k}^{(l+1)}\) must satisfy the conservation equation at time interval \(l\), that is:

\[
\sum_{k \in K_{i}^{(l+1)}} h_{k}^{(l+1)} = g_{i}^{(l+1)}
\]

Therefore:

\[
\begin{align*}
    h_{k}^{(l+1)} &= h_{k}^{(l)} - \frac{1}{l} g_{i}^{(l+1)} \quad \text{for } k, \tilde{k} \in K_{i}^{(l+1)} \\
    h_{\tilde{k}}^{(l+1)} &= h_{\tilde{k}}^{(l)} + \frac{|K_{i}^{(l+1)}| - 1}{l} g_{i}^{(l+1)}
\end{align*}
\]

(10)

**References**


M. Ben-Akiva and M. Bierlaire (1999), Discrete Choice Methods and Their Applications to Short Term Travel Decisions, in: Transportation Science Handbook, Kluwer


