Comparison of assignment methods for simulation-based dynamic-equilibrium traffic assignment

Michael Mahut [1,2], Michael Florian [2,1], Nicolas Tremblay [1]

[1] INRO Consultants Inc, Montreal, Canada
[2] Centre for Research on Transportation (CRT), University of Montreal, Canada

ABSTRACT

This paper reports on the evaluation of alternative algorithms for dynamic, or time-varying, equilibrium traffic assignment. The algorithms are used for pre-trip assignment, which reflects driver familiarity with expected traffic conditions, and are appropriate for off-line applications which require a more detailed analysis – such as temporal network flows and sensitivity to traffic control measures – and thus are not well served by static assignment models. The model is formulated as a time discrete variational inequality and a solution algorithm is developed. The determination of time-dependent path input flows is modeled as a master problem inspired from a simplicial decomposition approach. The determination of path travel times for a given set of path flows is the network-loading sub-problem, which is solved using the computationally efficient space-time queuing approach of Mahut. This solution method to the traffic flow problem is combined with heuristic driver behavior models to produce a fully-featured traffic simulation model. The complexity of the traffic simulation, which is motivated by a desire for a realistic representation of the system, results in an assignment map that is discontinuous and difficult to characterize analytically. Nevertheless, algorithms inspired from static network assignment have been found to work in practice for finding approximate dynamic equilibrium conditions on real-world networks of significant size. The routing algorithms are evaluated primarily on the proximity of the converged assignment to the dynamic equilibrium conditions. Other measures of algorithm performance include the speed of convergence (number of iterations), computational burden (per iteration), and computer memory requirements. The reported applications of this model include references to studies carried out in several cities around the world and some detailed model calibration results for the cities of Bakersfield, USA and Montreal, Canada. The results show that this model may be calibrated for medium-size networks and the dynamic flows obtained with very reasonable computing times.

***Index Terms* – dynamic traffic assignment, method of successive averages, projected gradient, traffic simulation, queuing models**

1. INTRODUCTION

This paper reports on the evaluation of alternative algorithms for dynamic equilibrium traffic assignment, which are used in conjunction with a detailed traffic simulation model. In order to provide a common terminology to the various models, it is convenient to refer to the main components of any dynamic traffic model: the route-choice mechanism, the determination of the path input flows and the network-loading mechanism. The latter is the method used to represent the evolution of the traffic flow over the links of the network once the route choice and the path input flows have been determined.

Some of the most popular dynamic traffic models today are those based on the representation of the behavior of each driver regarding car following, gap acceptance and lane choice. These are micro-simulation models. See for instance CORSIM (http://www.fhwa-tsis.com/corsim_page.htm), INTEGRATION (Van Aerde, 1999), AIMSUN2 (Barceló et al, 1994) (http://www.tss-bcn.com), VISSIM (http://www.ptv.de), PARAMICS (http://www.quadstone.com).See also DRACULA (http://www.its.leeds.ac.uk/software/dracula/)and MITSIM (Yang, 1997) (http://web.mit.edu/its/products.html) which are the results of academic research.

The successful and efficient use of micro-simulations is commonly limited to relatively small size networks. Their application has been hindered for medium-to-large networks by the relatively high computation time and the effort required for a proper model calibration.

The aim of handling larger networks with reasonable computational times has led to the development of so-called "mesoscopic" approaches to traffic simulation. The aim is to obtain a traffic representation that still captures the basic temporal congestion phenomena, but models the traffic dynamics with less fidelity. Some of the earliest examples of such an approach are CONTRAM (Leonard et. al., 1989) (www.contram.com) and SATURN (Van Vliet, 1982), which are commercially available packages that are still used in England and elsewhere. More recently the FHWA Dynamic Traffic Assignment Project has supported the development of DYNASMART (Mahmassani et al., 2001) and DYNAMIT (Ben-Akiva et al., 1998) (http://web.mit.edu/). Another approach to dynamic traffic assignment is that based on cellular automata theory (Nagel and Schreckenberg, 1992), which has been implemented in the TRANSIMS software (http://transims.tsasa.lanl.gov), developed by the Los Alamos National Laboratories in the USA.

Ziliaskopoulos and Lee (1997) developed a dynamic traffic assignment model based on the cell transmission model which is a particular solution method for the classical hydrodynamic traffic flow model (Lighthill and Whitham, 1955) (Richards, 1956). METACOR (Diakakis and Papageorgiou, 1996) and METANET (Messmer et al., 2000a are based have their roots in macroscopic traffic flow theory. These models are based on the work of Papageorgiou (1990) and Messmer et al., 2000b respectively. The network loading method is based on a second order (p.d.e.) traffic flow model. Another line of research is that of analytical dynamic traffic assignment models, which have their roots in the mathematical programming and optimal control approaches static network equilibrium models which are based on link travel time functions. There is a very large body of literature that contains academic contributions made by using this approach. A good reference is a special issue of *Networks and Spatial Economics* (Vol 1, Issue 3/4 , 2001). The paper by Friesz et al (1993) provides a formulation of an equilibrium dynamic traffic model which serves as the basis for the algorithm developed here.

The network loading method used in this work is based on a traffic simulation model that was designed to produce reasonably accurate results with a **minimum** number of parameters and a **minimum** of computational effort (Mahut, 2001, Astarita et al., 2001). The underlying structure of the network model and the car moving logic have more in common with microscopic than with mesoscopic approaches, as it is designed to capture the effects of car following, lane changing and gap acceptance. This method could be characterized as a *simplified* microscopic model, as it employs less complex variants of the car-following, lane-changing and gap-acceptance models implemented in micro-simulation software packages that are intended for more detailed traffic modeling.

The paper is structured as follows. The next section is dedicated to the exposition of the network and demand representation used in the model; then the formal statement of the model and its discretized version are given. The next sections are dedicated to the statement of the algorithms and to the description of the network loading method. Applications of the model, along with comparisons of the alternative assignment algorithms, are then given and some conclusions end the paper.

2. NETWORK REPRESENTATION, TRAFFIC CONTROL AND DEMAND

The physical network is defined by links and nodes. Each link is defined its length, number of lanes and free-flow speed. Additional lanes on intersection approaches, for left and right turns, bus stops, etc… are required and are appropriate to the fidelity of the traffic simulation model. Similarly, at each node, each turning movement is defined by the lanes on its upstream (incoming) and downstream (outgoing) links that are permitted for the movement, along with a maximum turning speed. Maximum (free-flow) speeds on links and turning movements, when combined with the physical parameters of vehicle length and driver response time, produce the well-known fundamental relationship of traffic flow for the car-following model used in the traffic simulator. As a result, per lane flow capacities, storage (density) capacities, and negative (backward moving) shock-wave speeds are all determined by the specification of the maximum speed, vehicle length, and driver response time.

The model does not use geometrical information such as intersection size and shape, or the radii of curvature of the turning movements. Each lane of a link, and each turning movement, can be restricted to a subset of the vehicle classes, permitting the modeling of HOV (high-occupancy vehicle) lanes, or reserved lanes for buses and/or taxis, etc. The model also permits the specification of detailed traffic control information such as (pre-timed) signal timing and ramp metering plans.

The demand is defined by a time-sliced O-D matrix for each vehicle class. Each vehicle class is comprised of one or more vehicle *types*, which are distinguished by the physical attributes of the vehicle *effective length* and the driver/vehicle *response time*, as discussed above.

3. DYNAMIC TRAFFIC ASSIGNMENT – THE MODEL

Two different approaches are commonly used to emulate the path choice behavior of drivers: dynamic assignment *en route* and dynamic *equilibrium* assignment. In this work, the approach taken is to seek an approximate solution to the dynamic equilibrium (pre-trip) conditions.

The path choices are modeled as decision variables governed by a user optimal principle where each driver seeks to minimize the used path travel time. All drivers have perfect access to information, which consists of the travel times on all paths (used and unused). The solution algorithm takes the form of an iterative procedure designed to converge to these conditions.

The solution approach adopted for solving the dynamic network equilibrium model (1)-(3) is based on a temporal discretization into periods $\tau = 1, 2, ..., \left| \frac{T_d}{\Delta t} \right|$, where Δt is the chosen duration of a departure-time interval. This results in a time discrete model.

The mathematical statement of a time discrete version of the dynamic equilibrium problem is in the space of path flows h_k^τ, for all paths *k* belonging to the set K_i for an origin-destination $i \in I$, at time *t*. The time-varying demands are denoted $g_i^\tau, i \in I, all\ \tau$ .The path flow rates in the feasible region Ω satisfy the conservation of flow and non-negativity constraints

$$\Omega^\tau = \{h_k^\tau : \sum_{k \in K_i} h_k^\tau = g_i^\tau, i \in I, all\ \tau\ ;\ h_k^\tau \geq 0, k_i, i \in I, all\ \tau\}, \quad (1)$$

and a temporal version of Wardrop's (1952) user optimal route choice results in the model::

$$h_k^\tau \in \Omega, u_i^\tau(t) = \min_{k \in K_i} \left\{ s_k^\tau(t) \right\}$$

$$s_k^\tau = u_i^\tau \ if\ h_k^\tau > 0$$

$$s_k^\tau \geq u_i^\tau \ if\ h_k^\tau = 0 \text{ for all } k \in k_i,\ i \in I, \tau = 1, 2, ..., \left|\frac{T_d}{\Delta t}\right| \qquad (2),$$

which can be shown to be equivalent to solving the discrete variational inequality.

$$\sum_\tau \sum_{k \in K} s_k^\tau (h^\tau)\,(h_k^\tau - h_k^\tau) \geq 0 \qquad (3),$$

where $K = \bigcup_{i \in I} k_i$ where h^τ is the vector of path flows (h_k^τ) for all *k* and τ.

The demonstration of existence and uniqueness of a solution to this model depends on the properties of the mapping ***s(h[g]),*** that is the dependence of link and path travel times on the path input flows and the dependence of the path input flows on the demands. Since the properties of this mapping are not easily verified due to the fact that it is the output of a simulation model and not an analytical transformation, no claims are made about the existence or the uniqueness of a solution. The equilibrium principle is used as a **guide** in computing an approximate solution of the time discrete variational inequality.

The next sections present an MSA-based solution algorithm to this problem, followed by an algorithm inspired by the projected gradient method. A heuristic method which allows the maximum step size to increase with departure time, which is applicable to both the MSA and quasi-gradient algorithms, is presented afterwards.

4. MSA-BASED ALGORITHM

The solution algorithm used here consists of two main components other than the computation of the temporal shortest paths: a method to determine a new set of time-dependent path input flows, given the experienced path travel times at the previous iteration, and a method to determine the actual link flows and travel times that result from a given set of path inflow rates.. The algorithm furthermore requires a set of initial path flows. The general structure of the algorithm is shown schematically in Figure 1.

The path input flows $h_k^\tau, k \in K$ are determined by a variant of the method of successive averages (MSA), which is applied to each O-D pair *I* and time interval τ . An initial feasible solution is computed by assigning the demand for each time period to a set of successive shortest paths. Starting at the second iteration, and up to a pre-specified maximum number of iterations, *N*, the time-dependent link travel times after each loading are used to determine a new set of dynamic shortest paths that are added to the current set of paths.
iteration $n, n \leq N$, the volume assigned as input flow to each path in the set is g_i^τ / n , $i \in I,\ all\ \tau$.
After that, for iteration $m, m > N$, only the shortest among *used* paths is identified and the path input flow rates are redistributed over the known paths.

If the flow of a particular path decreases below a small predetermined value then the path is dropped and its remaining flow is distributed to the other used paths. This heuristic approach is akin to the restricted simplicial decomposition algorithm of Lawgphonpanich and Hearn (1984) for the solution of the static network equilibrium model with fixed demand. The algorithm is summarized below.

EQUILIBRIUM DTA ALGORITHM

- *Step 0 Initialization (iteration counter $l=1$):*

  Compute temporal shortest paths based on free-flow travel times.
  Load the demands to obtain an initial solution;
  Update iteration counter: $l=l+1$.

- *Step 1 Reallocation of input flows to paths:*

  Step 1.1 If ($l \le N$)
  Compute a new dynamic shortest path.

  Assign to each path *k* the input flow $h_k^l(\tau)=\frac{g_i^\tau}{l}$

  Step 1.2 If ($l > N$)
  Identify the shortest among used paths.
  Redistribute the flows as follows:

$$h_k^l(\tau)=\begin{cases} h_k^{l-1}(\tau)\left(\frac{l-1}{l}\right)+\frac{g_i(\tau)}{l}, & \text{if } s_k^l(\tau)=u_k^l(\tau) \\ h_k^{l-1}(\tau)\left(\frac{l-1}{l}\right) & \text{otherwise} \end{cases} \qquad (7)$$

- *Step 2 Stopping rule:*
  If $l \le L \; or \; RGap \le \varepsilon \Rightarrow STOP$;

  otherwise return to Step 1

While no formal convergence proof can be given for this algorithm, since the network loading map does not have an analytical form, a measure of gap, inspired from that used in static network equilibrium models may be used for qualifying a given solution. It is the difference between the total travel time experienced and the total travel time that would have been experienced if all vehicles had the travel time (over each interval τ) equal to that of the current shortest path.

Hence a Relative Gap for each departure time interval τ may be computed as

$$R\ Gap^\tau(n)=\frac{\sum_{i\in I}\sum_{k\in k_i} h_k^\tau(n)s_k^\tau(n)-\sum_{i\in I} g_i^\tau\ u_i^\tau(n)}{\sum_{i\in I} g_i^\tau\ u_i^\tau(n)} \qquad (8),$$

where $u_i^\tau(n)$ are the lengths of the shortest paths at iteration *n*. A relative gap of zero would indicate a perfect dynamic user equilibrium flow. Clearly this is a fleeting goal to aim for with any dynamic traffic assignment.

It is very important to note that this model, even though its general formulation is very similar to flow based models, is in fact a discrete vehicle model. The network loading procedure, as realized by the event based simulation, moves **individual cars** on the links of the network. This is discussed in more detail in a later section.

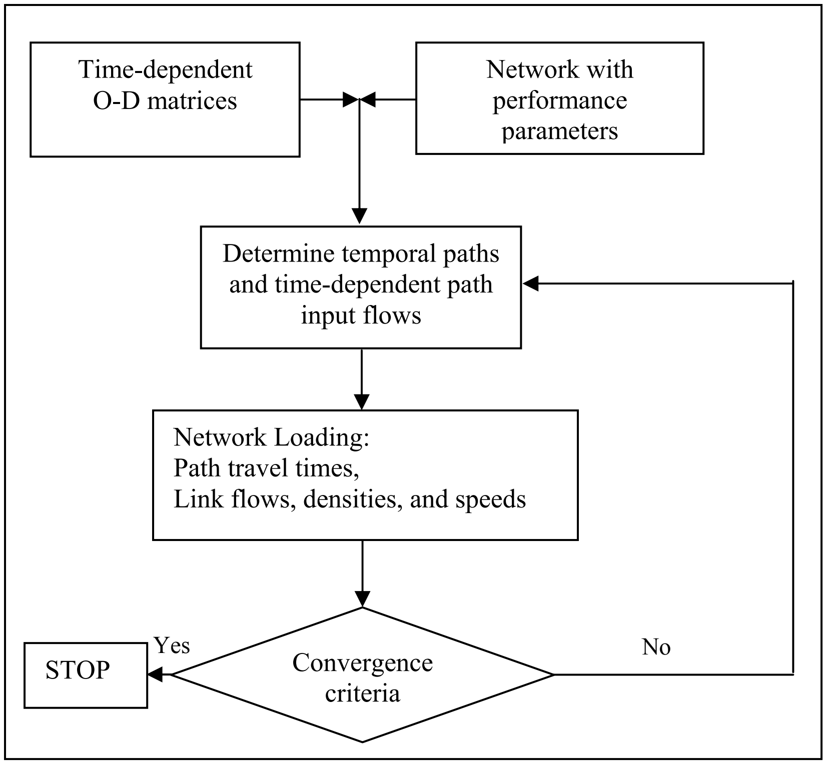

Figure 1. Structure of the solution algorithm for the DTA model

5. GRADIENT-BASED ALGORITHMS

The equilibration algorithms used in static equilibrium models that operate in the space of path flows provide some ideas that may be adapted heuristically for the solution of the dynamic equilibrium traffic assignment. These algorithms are adaptations of the classical convex simplex, projected gradient and reduced gradient algorithms implemented with a Jacobi or a Gauss Seidel decomposition scheme. Some selected references on the topic are Leventhal et al (1973), Dafermos (1971) and the text by Patriksson (1994).

The algorithm that is implemented for the solution of the DTA model considered in this paper operates in the space of path flows. Hence it is very attractive to adapt the equivalent of the projected gradient and the reduced gradient algorithms, even though there is no formal objective function that can be identified and the model formulation is a time discrete variational inequality. Since there is no objective function the step sizes adopted are those of the MSA method or the modified MSA method described below.

In order to state the algorithms (for one O-D pair) the notation used is the following. Let K^+ be the set of paths with positive flow. Let s_k is the cost (time) of a path and $\overline{s}$ be the average value of the path costs; p_k is the proportion of input flows to the paths $k \in K^+$; d_k Is the direction of change for each path and d_k^n is the normalized direction; α is the MSA step size.

The quasi projected gradient algorithm modifies the flow changes by using the following steps:

1. Compute the vector of $\bar{s} - s_k = d_k, \quad k \in K^+$;

2. Normalize the vector $d_k^n = \dfrac{\bar{s} - s_k}{\sum_k |d_k|}$;

3. Check for $\alpha_{\max}$, the largest value of α : $\alpha_{\max} = \max\left[p_k / d_k^n \middle| d_k^n < 0 \right]$ which would diminish the input proportion of a path to 0;

4. The step size is $\alpha = \min\left(\alpha_{\text{MSA}}, \alpha_{\max}\right)$;

5. Update the path proportions $p_k = p_k + \alpha d_k^n$.

The quasi reduced gradient algorithm modifies the flow changes by using the following steps:

1. Select the path that has the largest flow: $k^* = \arg\max\left\{h_k\right\}$;
2. Compute the vector of $d_k = \left(s_k^* - s_k\right), k \neq k^*$ for all k not equal to k^* and $d_k^* = -\sum_{k \neq k^*} d_k$;
3. Normalize as above $d_k^n = \dfrac{\bar{s} - s_k}{\sum_k |d_k|}$;
4. Check for $\alpha_{\max}$, the largest value of alpha: $\alpha_{\max} = \max\left[\dfrac{p_k}{d_k^n} \middle| d_k^n < 0 \right]$;
5. The step size is $\alpha = \min(\alpha_{\text{MSA}}, \alpha_{\max})$;

6. Update the path proportions $p_k = p_k + \alpha d_k^n$.

The trips that are assigned to be loaded on each path are simply the product of the demand for the O-D pair multiplied by the path proportions. Then the simulation that achieves the network loading is carried out.

6. TIME-VARYING STEP-SIZE ADJUSTMENT

A basic observation about the behavior of the MSA algorithm is that the assignment for a specific departure-time interval is further away from the equilibrium conditions (measured by the *relative gap* in this work) with later departure times. In fact, for a majority of the real-world networks that have been tested to date, the relationship between departure time and relative gap is monotonically non-decreasing (exceptions to this rule are invariably cases with very severe congestion). Another consistent trend in time-varying relative gap (see Figure 5, below) is that later departure-time intervals require more iterations before converging to a stable value of relative gap: again, the relationship is very consistent across various networks.

One basic explanation for these phenomena is that the travel times of later-departing vehicles are affected by earlier-departing vehicles, and thus the convergence for a later-departing interval cannot be achieved until it has first been achieved for the prior interval. This inherent property of the model suggested the possibility that the higher values of relative gap in the later-departing intervals might be partially a result of the fact that the MSA step-size is the same for all departure-time intervals at each iteration. To put it simply, by the time (in iterations) that a later interval finally starts to converge, the step-size is so small that not enough flow is being moved away from the longer paths towards the shortest path. Another reason for the increasing values of relative gap is simply that the later-departing vehicles incur higher congestion, and that it is more "difficult" for the algorithm to reach equilibrium conditions as congestion increases. This idea is also supported by the various scenarios that have been tested to date, which cover a wide range of congestion levels.

These observations are the basis of a time-varying step-size heuristic, which gradually modifies the step-sizes applied to later departure intervals. The heuristic uses an integer *reset* parameter n, and is first applied in step 2.2 of the MSA algorithm above (step 2.1 remains unchanged). The first $n*d$ iterations, where

$$d = \left|\frac{T_d}{\Delta t}\right|$$

is the number of departure-time intervals, of step 2.2 are a transitory period during which the l-values of a departure intervals are iteratively rolled back using a cascading pattern, which can be succinctly described as follows.

> Modified step 2.2: at iteration $N+n*r$, $1<r\leq d$; apply $l=l-n$ for departure intervals $r, r+1, ..., d$.

This rule creates a cascade or staircase pattern of *l-values*, when viewed in a table of iteration vs. departure interval, that is gradually built during iterations $N+n, N+n+1, ... N+n*d$. The l-values for the first departure interval are not modified using this heuristic. After iteration $N+n*d$, the l-values for increasing departure intervals are offset by exactly n and this pattern remains until termination of the DTA algorithm.

7. NETWORK LOADING MODEL

The core of the traffic simulation model is the following simplified car-following model:

$$x_f(t) = \min\left[x_f(t-\varepsilon) + \varepsilon \cdot V, x_l(t-R) - L\right],$$

Where *x(t)* is the trajectory of a vehicle (position as a function of time), *L* is the effective vehicle length, *R* is the driver/vehicle response time, *V* is the free flow speed, and ε is an arbitrarily short time interval. The subscripts *f* and *l* denote the trajectories of the following and leading vehicles, respectively. This model considers only the free-flow speed (without acceleration constraints) combined with a simple collision avoidance model, and can easily be shown to yield the well-known "triangular" relationship between traffic flow and density, also known as the fundamental diagram of traffic (Mahut, 2000). This relationship can be solved in a rigorous way, while only calculating the time at which a vehicle enters and exits each link, using the following expression:

$$t_n(0) = \max\left[t_n(-X_1) + \frac{X_1}{V_1},\ t_{n-1}(0) + \left(R + \frac{L}{V_2}\right),\ t_{n-X_2/L}(X_2) + R\left(\frac{X_2}{L}\right)\right]$$

Where X_1 and X_2 are the lengths of the links upstream and downstream of the position x=0, V_1 and V_2 are the free-flow speeds of these two links, respectively, and the subscripts indicate vehicle numbering in sequential order. This "link-based" solution provides a very practical and computationally efficient way to model traffic without actually calculating the state variables (position, speed, etc..) of each vehicle at each second or less (using a *time-step* solution). It should be noted that this expression only applies strictly to the case of a one-lane link.

A multi-lane version of the above relationship maintains the same property of only calculating the entrance and exit times of each vehicle, but also captures the interactions between vehicles due to lane-changing maneuvers (Mahut 2001). The multi-lane model also employs a complex set of heuristics for modeling a driver's lane-selection decisions, which takes into consideration the driver's intended path downstream of his current position, the lane(s) that may be used to execute the next turning movement, and the prevailing traffic conditions on each lane between the driver and the end of the link. This model is described in more detail elsewhere in the literature (Mahut 2001), (Florian et al 2005), (Tian et al, 2006).

The link-based expression above determines the earliest entrance time for any vehicle wishing to enter the link, as a function of the historical record of exit times and prior entrance times. The information is used in a way that rigorously applies the above car following model. Essentially, the car following model is extended in a recursive way to apply over a sequence of vehicles. For example, rather than modeling the relationship between vehicle 1 and vehicle 2 (higher number is follower of lower number), and the subsequent relationship between vehicles 2 and 3, the model allows the impact of vehicle 1 on vehicle 3 to be expressed directly. This can be extended to any number of vehicles, which is the essence of the link-based expression above. Conceptually, as the primary source of delays in a traffic network is at the nodes, the role of (link) traffic dynamics in such a model is to correctly propagate the delays incurred by vehicles at the downstream end of the link to the upstream end, where these delays (congestion spill-back) will affect the entrance times of vehicles to the link, i.e. the maximum in-flow rate. Rather than explicitly modeling the position of each vehicle in order to determine when link congestion begins to affect these entrance times, the exit delays are propagated directly back to the entrance. This particular characteristic – the ability to rigorously solve the traffic model over entire links – has also been demonstrated for the kinematic wave model based on the triangular fundamental diagram (Newell, 1992).

This model of traffic dynamics – which can be characterized as a continuous time, continuous space, discrete-flow model – is combined with a node model which explicitly represents the traffic control system (pre-timed traffic signals), and also models driver interactions at uncontrolled points of conflict with gap-acceptance logic. The combined system is then solved using a discrete-event (*event based*) algorithm which allows entire networks to be modeled. Event-based models are fundamentally different from time-step models, in the following way:

- Time-step model: at each time step (usually 1 second or less), for each vehicle, all input data required for all of the individual models (car-following, lane-changing, gap-acceptance, route choice), is updated and the outputs (acceleration, deceleration, lane changing action, etc…) are re-computed.
- Event-based model: the individual models are only re-executed at the moment in time at which any of their relevant inputs change. The models are thus designed in such a way that their outputs (e.g. vehicle speed) will remain valid for as long as the inputs to the calculation do not change. An event is generated at a specific point in (continuous) time, to reflect a change in information (stimuli) which must be acted upon. Generally, the execution of an event results in the creation of one or more subsequent events.

The result in this case is an event-based model that is very efficient computationally, but that also respects all of the basic laws of traffic flow and explicitly represents the mechanisms of congestion that occur in real traffic. As can be seen in the expression above, the car following model is a relatively simple one: this simplification (relative to other car following models in the literature) is necessary in order to permit an efficient event-based procedure. However, this results in a major savings in computation time relative to time-step simulation models (between one and two orders of magnitude). The computation time is particularly important in the context of a DTA model which must perform numerous iterations of the simulation model in the process of solving for the equilibrium assignment. The fact that DTA modeling tends to be done on larger networks, where there is significant route choice, only makes the computation time issue even more critical.

An overview of the properties the simulation model is as follows:

- Individual vehicle movements are modeled, resulting in a realistic representation of congestion mechanisms (mainly at nodes), and the resulting spill-back (propagation) of congestion across lanes (laterally) and to upstream links (longitudinally);
- Advanced lane-selection rules that employ *look-ahead* logic; the latter is not restrained to a fixed number of links downstream of the driver;
- A deadlock prevention algorithm identifies cycles of links that are very close to locking up (during the simulation itself), and manages the inflows to these cycles, much like an adaptive traffic control system, in order to prevent deadlock and maintain traffic flow;
- Very fast computation times compared to time-step simulation models;

8. APPLICATIONS

This dynamic traffic assignment algorithm was coded in C++ using an object-oriented approach. The code runs under the Windows XP/2000 operating systems. The model was first applied in Stockholm and then in Calgary, Canada. The Calgary application and model calibration was reported in Mahut et al (2004).

Other applications have been accomplished in Basel (Switzerland), Bakersfield (California, U.S.A.), Montreal (Canada) and elsewhere. We give below the relative sizes of some of these applications and the computer resources on a 1.6 Ghz Intel Centrino notebook computer operating under Win XP.

City	**Zones**	**Nodes**	**Links**	**Vehicles**	**Area (km^2)**	**RAM (MB)**	**CPU/iteration**
Calgary	76	402	752	12,563	14		6.4 s
Bakersfield	64	368	582	47,023	45	75 M	51.2 s
Basel	120	576	1450	65,631	21		50.3 s
Stockholm	114	1191	2080	108,000	200	90	1.5 min
Montreal	232	966	2563	232,667	45	550	4.5 min

We present next some details of the Montreal application. The static planning model for the Montreal Region consists of 1,400 zones, 15,000 nodes and 33,500 links. A sub-area identified in pink in Figure 2 indicates the sub-area that was identified for the study of an improved urban arterial facility. A traversal matrix was computed for the sub-area and then adjusted by using link counts. More detail was added where necessary and some 320 traffic signals were coded by the staff of the City of Montreal. The resulting model has 232 zones, 966 nodes, 2,563 links and the total vehicular demand is 232,667. The data collection, coding and calibration effort took some nine months.

Figure 3 shows a snapshot of a snapshot of link-based simulation results for a 5-minute time interval: link outflow is shown as bar width, while level of congestion (0-100% scale) is indicated with color. A typical convergence plot for 10 departure-time intervals for the 6:30 – 9:00 AM peak hour are shown in Figure 4. The calibration results, displayed as a scatterplot of traffic counts vs. model outputs as shown in Figure 5, were found to be excellent. Figure 6 shows a comparison of simulation results (link outflow and congestion) for before and after the capacity is increased on the Notre-Dame facility (indicated with an arrow). The wider bars and cooler colors indicate higher flows and lower congestion after the facility is improved.

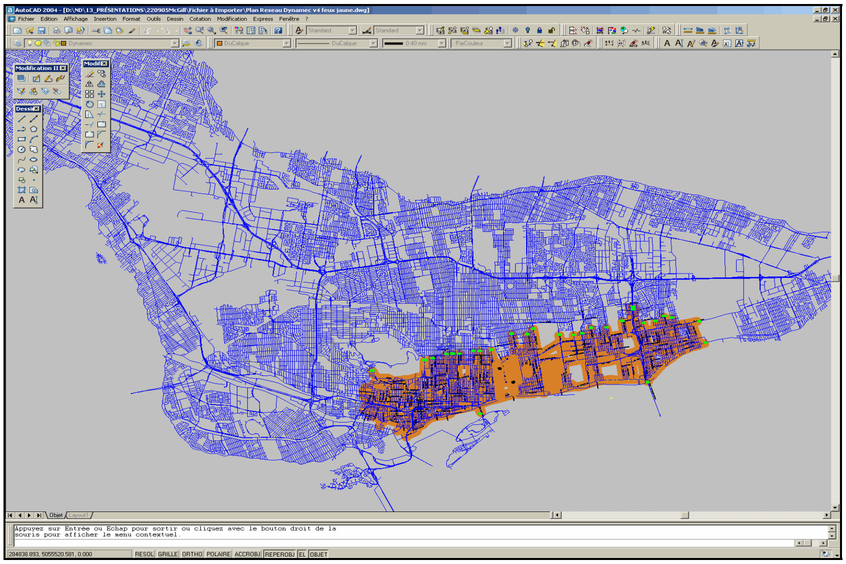

Figure 2. The Notre Dame highway sub-area

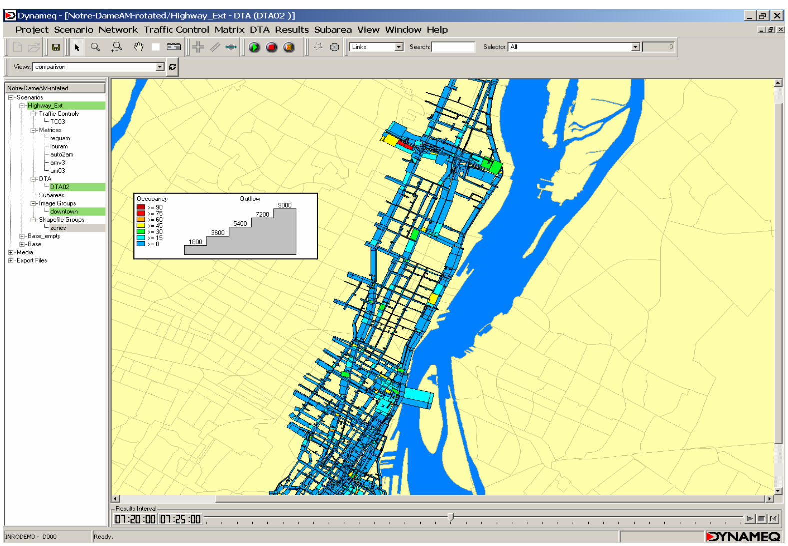

Figure 3. Sub-area AM flows and congestion levels

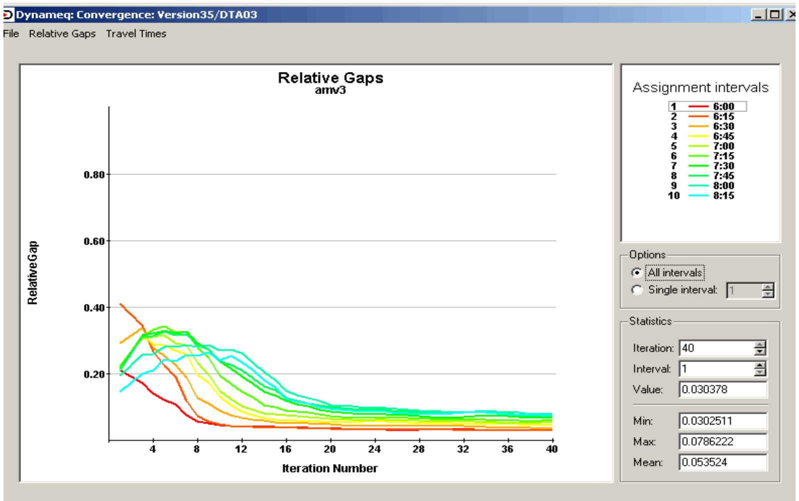

Figure 4. Convergence for AM peak period (3 hours)

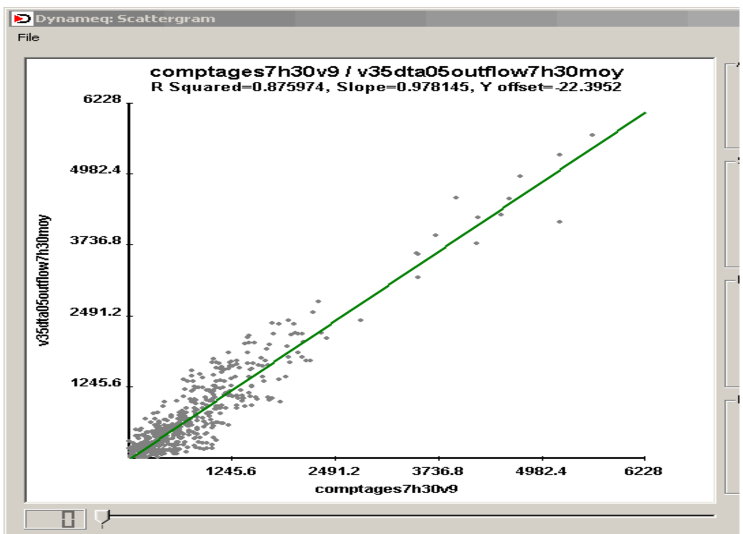

Figure 5. Calibration results: predicted vs. observed flows

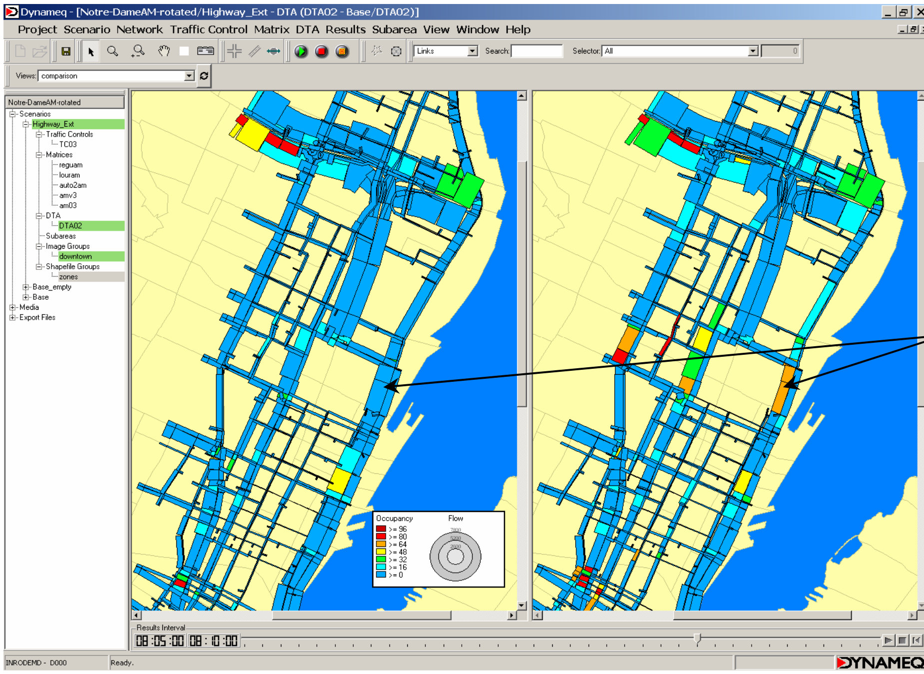

Figure 6. Scenario comparison: left window after improvement.

9. COMPARISON OF ASSIGNMENT ALGORITHMS

The section compares convergence results for the MSA algorithm and quasi projected gradient method with time-varying step-size adjustment. The quasi reduced gradient method was not found to perform well in the tests and thus this method was removed from the evaluation.

The methods were first compared on the Bakersfield network, which appears in the table of the previous section and is shown in Figure 7. This application is smaller than the Montreal network and focuses on a 7-mile (11.2 km) section of Interstate freeway and part of the urban arterial network. The sub-area considered was obtained from the Kern COG regional static network and the origin destination matrices for private cars and trucks were synthesized by using a combination of O-D matrix adjustment techniques and new counts.

Plots of relative gap (for six departure-time intervals) are shown for the MSA algorithm in Figure 8. Figure 9 shows the relative gaps for the MSA algortihm with time-varying step-size adjustment, while Figure 10 shows the relative gaps for the quasi projected gradient with step-size adjustment. The offset parameter (n) used for the step-size adjustment was 3.

Looking at figures 8 to 10, it can be seen that the step-size adjustment rule, when added to the MSA algorithm, significantly accelerated the convergence. The quasi-projected gradient, in combination with the step-size adjustment rule, further accelerated the convergence and also resulted in a smoother overall convergence. Although a detailed investigation is beyond the scope of this paper, it has been observed that the relative gaps do not ultimately converge to zero, regardless of how many iterations are executed. Rather, for a given network and demand, there is a stable value of relative gap for each interval to which system converges, with more or less fluctuation, beyond which no more improvement is observed. The improved methods were found to consistently lower these stable gap measures, and thus provided *tigher* convergence as well as *faster* convergence.

Visually, this stability can be clearly observed in Figure 10, which shows the gap measures for the quasi projected gradient with step-size adjustment. There is no significant improvement beyond roughly iteration 45, where the average relative gap is 3.2%. For practical applications, this is an entirely acceptable approximation of dynamic equilibrium, indicating that more iterations would not be required. By comparison, the average relative gaps at iteration 45 of the MSA algorithm and MSA with step-size adjustment are 6.6% and 3.9%, respectively. Even if these values might also be considered adequate, the fact that the gaps are still significantly improving at iteration 45 in Figures 8 and 9 makes it impractical, or at least unsatisfying, to stop the the algorithms at this point. The reduction in the number of iterations is at least as significant, for practical purposes, as the tigher convergence obtained once the gap values stabilize.

It must be stressed that due to the complexity of the traffic simulation model, there are practical limitations as to how close the algorithms can get the system to perfect equilibrium. For example, the reductions obtained in total CPU (number of iterations) and in the relative gap values, are both roughly 50% (comparing MSA to quasi projected gradient with step size adjustment). These improvements may seem modest in the realm of static assignment methods, where improvements in convergence may be measured on a log scale. However, in the realm of simulation-based assignment, and in particular for a high-fidelity simulation model as is used here, these improvements are remarkable.

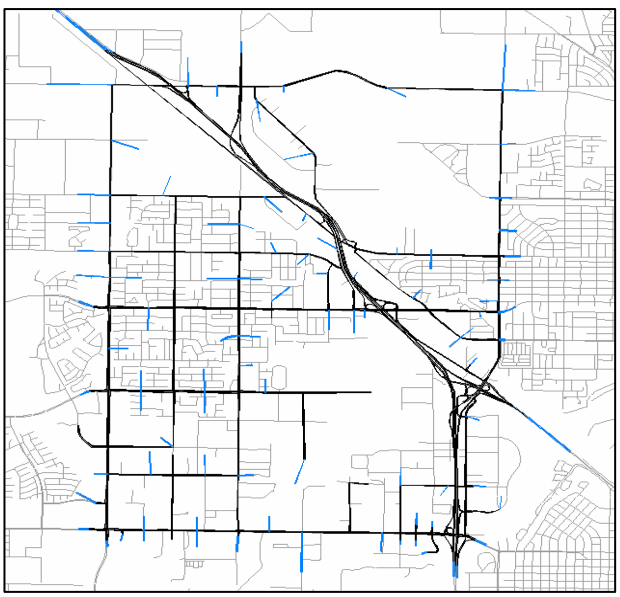

Figure 7. Bakersfield Network.

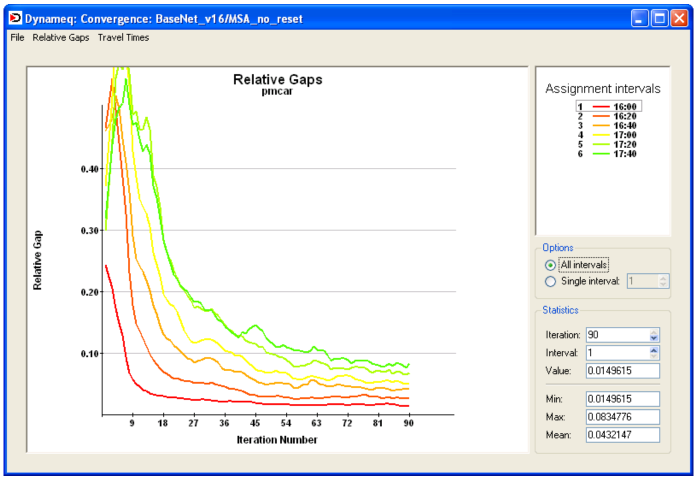

Figure 8. Relative Gaps for MSA algorithm.

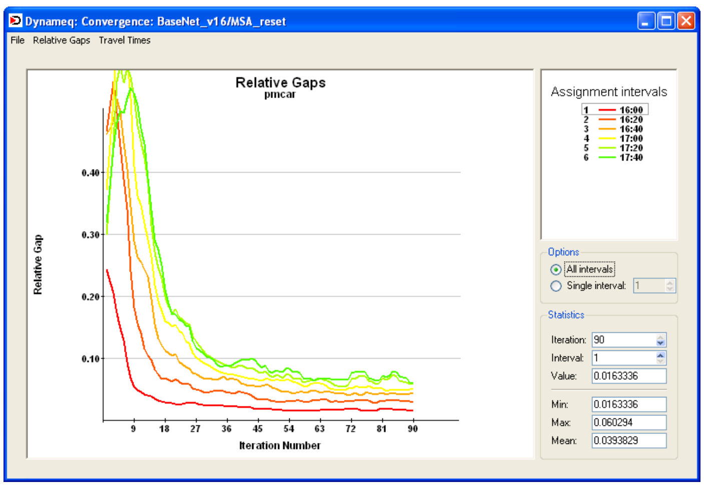

Figure 9. Relative Gaps for MSA with time-varying step-size rule.

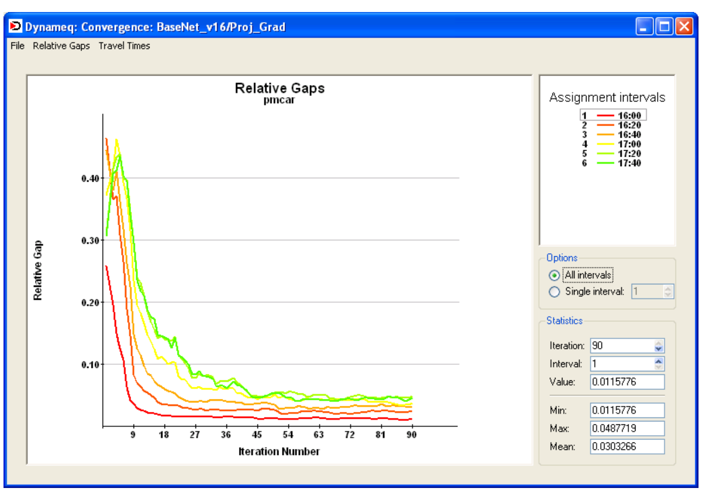

Figure 10. Relative Gaps for quasi projected gradient with time-varying step-size rule.

The algorithms were also tested on a more recent version of the Montreal sub-area which is more than double the size (in area) of the one shown in Figure 2. This network was run on a laptop computer with a Centrino-2 dual core 2.3 GHz processor. The scenario and performance measures for this test are as follows.

City	Zones	Nodes	Links	Vehicles	Area (km^2)	RAM (MB)	CPU/iteration
Montreal (2)	401	2216	6488	320,742	115	950	241 s

The general trend towards faster convergence is again visible to the eye. The same trend towards tighter convergence is also found, although is less dramatic, with average gap measures of 3.9%, 3.3% and 2.8% for the MSA, MSA with step-size adjustment, and quasi projected gradient with step-size adjustment, respectively (see Figures 11 to 13). This network exhibited more fluctuation in the gap measures, particularly for the later departure intervals. This appears to be due to acute congestion areas in the network, in which congestion increases rapidly even with very small changes in path flows.

The CPU times for the assignment methods were also investigated, but since the assignment algorithm only contributes about 1% of the total (the traffic simulator contributes the remaining 99%), modest changes in CPU time among the assignment alogrithms is of no practical concern. For the large Montreal network, the quasi-projected gradient with step-size adjustment required about twice the CPU time as for the regular MSA algorithm (0.6 vs. 0.3 seconds per iteration).

10. CONCLUSIONS

Two assignment algorithms, namely MSA and a quasi projected gradient method, along with a time-varying step-size adjustment heuristic, were tested in a simulation-based dynamic equilibrium model. The primary measure of performance was the relative gap, which gives a direct indication of how closely the algorithms are able to bring the simulated system to satifying dynamic equilbrium conditions. The best performace was consistenly obtained with the combination of the quasi projected gradient and the time-varying step-size adjustment heuristic. Compared to the MSA algorithm this method provided considerably faster convergence, which for a typical network allowed the algorithm to achieve practical convergence in half as many iterations.

The use of a microscopic simulator for mapping the path input flows to time-varying experienced path travel times imposes practical limitations on the degree of convergence that can possibly be attained. Nevertheless, average relative gaps (across departure intervals) of 3% were typically achieved with a reasonable amount of computing time, which for practical applications is certainly an acceptable approximation of dynamic equilbrium conditions. The combination of a high fidelity representation of the traffic system on a medium-sized network (6500 links in this case), with good equilibrium properties and reasonable RAM and CPU requirements, is unprecedented.

The overall method has excellent potential for use in practice for a variety of off-line applications for short-term planning, such as infrastructure planning, contruction staging, work-zone analysis, evacuation modeling, and evaluation of alternative traffic control plans. The model is also appropriate for longer-term planning studies, which in some cases also required a higher level of detail and realism in the representation of the traffic system.

***Acknowledgement* – This work was partially sponsored by a grant Fellowship of the Natural Sciences and Engineering Council of Canada (NSERC). We acknowledge the permission of the City of Montreal and California Department of Transportation to include material that is based on the work of their staff in this paper.**

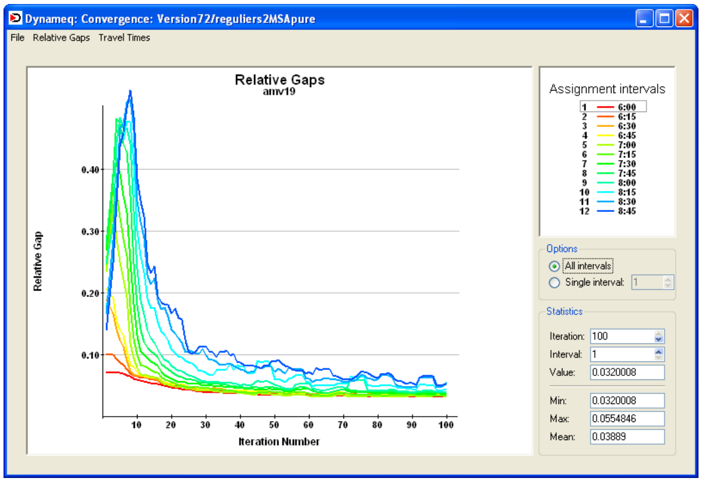

Figure 11. Relative Gaps for MSA algorithm.

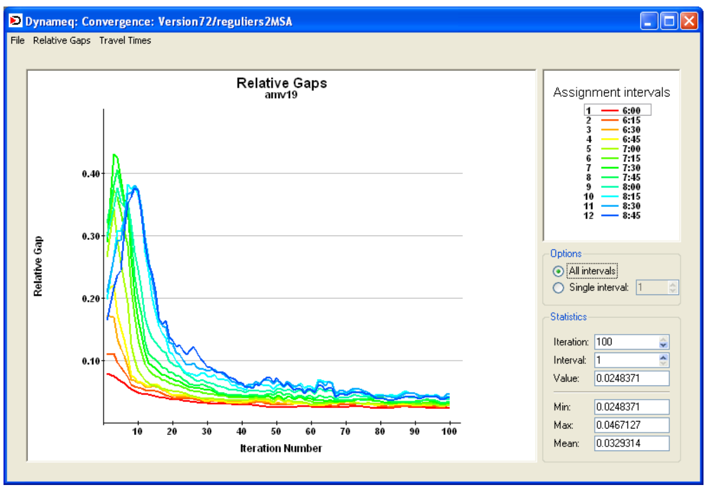

Figure 12. Relative Gaps for MSA with time-varying step-size rule.

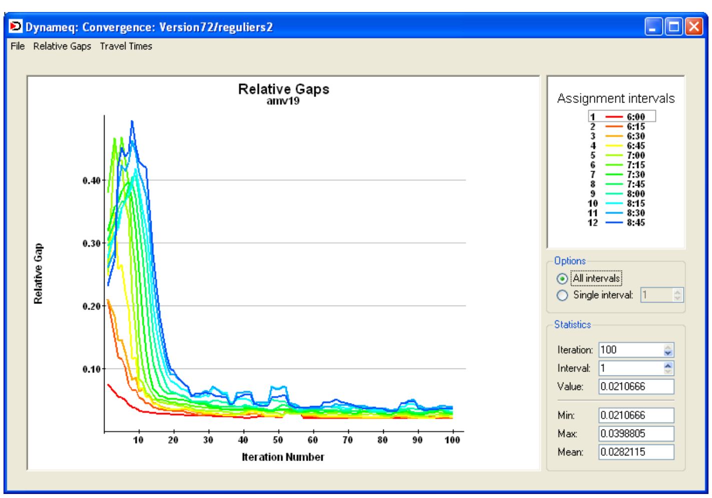

Figure 13. Relative Gaps for quasi projected gradient with time-varying step-size rule.

REFERENCES

Astarita, V., Er-Rafia, K., Florian, M., Mahut, M., Velan, S. (2001). Comparison of Three Methods for Dynamic Network Loading. *Transportation Research Record*, 1771:179-190.

Barceló, J. (2000). The Role of Traffic Simulation in Advanced Traffic Management Systems. Presented at the Spring meeting of INFORMS, Salt Lake City, USA. May 7-10, 2000.

Barceló, J., Ferrer, J.L., and Grau, R. (1994). AIMSUN2 and the GETRAM Simulation Environment. *Internal Report*, Departamento de Estadistica e Investigacion Operativa. Universitat Politecnica de Catalunya. See also http://www.tss-bcn.com.

Ben-Akiva, M., Koutsopoulis, H., Toledo, T. (2000). MITSIMLab: Recent Developments & Applications. Presented at the Spring meeting of INFORMS, Salt Lake City, USA. May 7-10, 2000.

Ben-Akiva, M., Koutsopoulos, H.N., Mishalani, R. (1998). "DynaMIT: A Simulation-Based System for Traffic Prediction". Paper presented at the DACCORD Short Term Forecasting Workshop, Delft, The Netherlands. See also its.mit.edu.

Dafermos,S.C. (1971) An extended traffic assignment model with application to two-way traffic. *Transportation Science* 5, 366-389

Diakaki, C., and Papageorgiou, M. (1996). Integrated Modeling and Control of Corridor Traffic Networks using the METACOR Modeling Tool. Dynamic Systems and Simulation Laboratory, Technical University of Crete. Internal Report No. 1996-8. Chania, Greece. pp. 41.

Florian, M., Mahut, M. and Tremblay, N. (2001). A Hybrid Optimization-Mesoscopic Simulation Dynamic Traffic Assignment Model. IEEE Intelligent Transportation Systems Conference Proceedings. Oakland, California, USA. August 25-29, 2001.

Friesz, T., Bernstein, D., Smith, T., Tobin, R., Wie, B. (1993). A variational inequality formulation of the dynamic network user equilibrium problem. *Operations Research*, 41:179-191.

Lawphongpanich, S. and Hearn, D.W. (1984). Simplicial Decomposition of the Asymmetric Traffic Assignment Problem. Transportation Research B, 17: 123-133

Lighthill, M.J. and Whitham, G.B. (1955). On kinematic waves I: Flood movement in long rivers. II: A theory of traffic flow on long crowded roads. *Proceedings of the Royal Society of London*, A229:281-345.

Leonard, D.R., Gower, P. and Taylor, N.B. (1989). CONTRAM: Structure of the Model. *Transport and Road Research Laboratory (TRRL) Research Report 178*. Department of Transport, Crowthorne. See also http://www.contram.com/.

Leventhal,T., Nemhauser, G., Trooter,L.Jr. 91973). A column generation algorithm for optimal traffic assignment, *Transportation Science* 7, 168-176.

Mahmassani, H.S., Abdelghany, A.F., Huynh, N., Zhou, X., Chiu, Y.-C. and Abdelghany, K.F. (2001). DYNASMART-P (version 0.926) User's Guide. Technical Report STO67-85-PIII, Center for Transportation Research, University of Texas at Austin.

Mahut, M. (2001). Discrete flow model for dynamic network loading. Ph.D. Thesis, Département d'informatique et de recherhe opérationelle, Université de Montréal. Published by the Center for Research on Transportation (CRT), University of Montreal.

Mahut, M., Florian, M., Tremblay, N., Campbell, M., Patman, D. and McDaniel., Z. (2004) Calibration And Application Of A Simulation-Based Dynamic Traffic Assignment Model, *Transportation Research Record* 1876: 101-111.

Messmer, A. (2000a). METANET A Simulation Program for Motorway Networks (Documentation). Dynamic Systems and Simulation Laboratory, Technical University of Crete. Chania, Greece.
Messmer, A (2000b). METANET-DTA An Exact Dynamic Traffic Assignment Tool Based on METANET. Dynamic Systems and Simulation Laboratory, Technical University of Crete. Chania, Greece. pp. 37.

Nagel, K. and Schreckenberg, M. (1992). A cellular automaton model for freeway traffic. *Journal de Physique I France*, 2:2221-2229.

Newell, G.F. (1993). A simplified theory of kinematic waves in highway traffic. Part I: General Theory. *Transportation Research B*, 27B(4):281-313.

Papageourgiou, M. (1990). Dynamic Modelling, Assignment and Route Guidance in Traffic Networks. Transportation Research, 24B(6), 471-95.

Patriksson, M. (1994), *The Traffic Assignment Problem: Models and Methods. Topics in Transportation,* VSP BV, Utrecht, the Netherlands.

Richards, P.I. (1956). Shock waves on the highway. *Operations Research*, 4:42-51.

Van Aerde, M. (1999). INTEGRATION Release 2.20 for Windows: User's Guide. MVA and Assoicates, Kingston, Canada.

Van Vliet, D (1982). SATURN - A Modern Assignment Model. In: Traffic Engineering and Control, 23, pp 578-581.

Wardrop, J. G. (1952). Some theoretical aspects of road traffic research, *Proceedings, Institute of Civil Engineers*, PART II, Vol.1, pp. 325-378.

Yang, Q. (1997). A Simulation Laboratory for Evaluation of Dynamic Traffic management Systems. Ph.D. Thesis, Massachusetts Institute of Technology. See also http://web.mit.edu/its/products.html.

Ziliaskopulos, A.K. and Lee, S. (1997). A Cell Transmission Based Assignment-Simulation Model for Integrated Freeway/Surface Street Systems. *Transportation Research Record* 1701, 12-2